Product ID: AA24-326A

November 21, 2024

Authored by:

Enhancing Cyber Resilience: Insights from CISA Red Team Assessment of a U.S. Critical Infrastructure Sector Organization

Executive Summary

The Cybersecurity and Infrastructure Security Agency (CISA) conducted a red team assessment (RTA) at the request of a critical infrastructure organization. During RTAs, CISA's red team simulates real-world malicious cyber operations to assess an organization's cybersecurity detection and response capabilities. In coordination with the assessed organization, CISA is releasing this Cybersecurity Advisory to detail the red team's activity—including their tactics, techniques, and procedures (TTPs) and associated network defense activity. Additionally, the advisory contains lessons learned and key findings from the assessment to provide recommendations to network defenders and software manufacturers for improving their organizations' and customers' cybersecurity posture.

Within this assessment, the red team (also referred to as 'the team') gained initial access through a web shell left from a third party's previous security assessment. The red team proceeded to move through the demilitarized zone (DMZ) and into the network to fully compromise the organization's domain and several sensitive business system (SBS) targets. The assessed organization discovered evidence of the red team's initial activity but failed to act promptly regarding the malicious network traffic through its DMZ or challenge much of the red team's presence in the organization's Windows environment.

The red team was able to compromise the domain and SBSs of the organization as it lacked sufficient controls to detect and respond to their activities. The red team's findings illuminate lessons learned for network defenders and software manufacturers about how to respond to and reduce risk.

- **Lesson Learned: The assessed organization had insufficient technical controls to prevent and detect malicious activity.** The organization relied too heavily on host-based endpoint detection and response (EDR) solutions and did not implement sufficient network layer protections.

- **Lesson Learned: The organization's staff require continuous training, support, and resources to implement secure software configurations and detect malicious activity.** Staff need to continuously enhance their technical competency, gain additional institutional knowledge of their systems, and ensure they are provided sufficient resources by management to have the conditions to succeed in protecting their networks.
- **Lesson Learned: The organization's leadership minimized the business risk of known attack vectors for the organization.** Leadership deprioritized the treatment of a vulnerability their own cybersecurity team identified, and in their risk-based decision-making, miscalculated the potential impact and likelihood of its exploitation.

To reduce risk of similar malicious cyber activity, CISA encourages critical infrastructure organizations to apply the recommendations in the **Mitigations** section of this advisory to ensure security processes and procedures are up to date, effective, and enable timely detection and mitigation of malicious activity.

This document illustrates the outsized burden and costs of compensating for insecure software and hardware borne by critical infrastructure owners and operators. The expectation that owners and operators should maintain the requisite sophisticated cyber defense skills creates undue risk. Technology manufacturers must assume responsibility for product security. Recognizing that insecure software contributes to these identified issues, CISA urges software manufacturers to embrace Secure by Design principles and implement the recommendations in the Mitigations section of this advisory, including those listed below:

- **Embed security into product architecture throughout the entire software development lifecycle** (SDLC).
- **Eliminate default passwords**.
- **Mandate MFA**, ideally phishing-resistant MFA, for privileged users and make MFA a default, rather than opt-in, feature.

Table of Contents

Introduction

CISA has authority to—upon request—provide analyses, expertise, and other technical assistance to critical infrastructure owners and operators and provide operational and timely technical assistance to federal and non-federal entities with respect to cybersecurity risks. (See generally 6 U.S.C. §§ 652[c][5], 659[c][6]). The target organization for this assessment was a critical infrastructure organization in the United States. After receiving a request for an RTA from the organization and coordinating the high-level details of the engagement, CISA conducted the RTA over approximately a three-month period.

During RTAs, a CISA red team simulates real-world threat actors to assess an organization's cybersecurity detection and response capabilities. During Phase I, the red team attempts to gain and maintain persistent access to an organization's enterprise network, avoid detection, evade defenses, and access SBSs. During Phase II, the red team attempts to trigger a security response from the organization's people, processes, and/or technology.

Drafted in coordination with the assessed organization, this advisory details the red team's activity and TTPs, associated network defense activity, and lessons learned to provide network defenders with recommendations for improving an organization's cybersecurity posture. The advisory also provides recommendations for software manufacturers to harden their customer networks against malicious activity and reduce the likelihood of domain compromise.

Technical Details

Note: This advisory uses the MITRE ATT&CK® Matrix for Enterprise framework, version 16. See Appendix: MITRE ATT&CK Tactics and Techniques for a table of the red team's activity mapped to MITRE ATT&CK tactics and techniques.

Phase I: Red Team Cyber Threat Activity

Overview

The CISA red team operated without prior knowledge of the organization's technology assets and began the assessment by conducting open source research on the target organization to gain information about its network [T1590], defensive tools [T1590.006], and employees [T1589.003]. The red team designed spearphishing campaigns [T1566] tailored to employees most likely to communicate with external parties. The phishing attempts were ultimately unsuccessful—targets ran the payloads [T1204], but their execution did not result in the red team gaining access into the network.

After the failed spearphishing campaigns, the red team continued external reconnaissance of the network [T1595] and discovered a web shell [T1505.003] left from a previous Vulnerability Disclosure Program (VDP). The red team used this for initial access [TA0001] and immediately reported it to the organization's trusted agents (TAs). The red team leveraged that access to escalate privileges [TA0004] on the host, discover credential material on a misconfigured Network File System (NFS) share [T1552.001], and move from a DMZ to the internal network [TA0008].

With access to the internal network, the red team gained further access to several SBSs. The red team leveraged a certificate for client authentication [T1649] they discovered on the NFS share to compromise a

system configured for `Unconstrained Delegation`. This allowed the red team to acquire a ticket granting ticket (TGT) for a domain controller [T1558.001], used to further compromise the domain. The red team leveraged this level of access to exploit SBS targets provided by the organization's TAs.

The assessed organization detected much of the red team's activity in their Linux infrastructure after CISA alerted them via other channels to the vulnerability the red team used for initial access. Once given an official notification of a vulnerability, the organization's network defenders began mitigating the vulnerability. Network defenders removed the site hosting the web shell from the public internet but did not take the server itself offline. A week later, network defenders officially declared an incident once they determined the web shell was used to breach the internal network. For several weeks, network defenders terminated much of the red team's access until the team maintained implants on only four hosts. Network defenders successfully delayed the red team from accessing many SBSs that required additional positioning, forcing the red team to spend time refortifying their access in the network. Despite these actions, the red team was still able to access a subset of SBSs. Eventually, the red team and TAs decided that the network defenders would stand down to allow the red team to continue its operations in a monitoring mode. In monitoring mode, network defenders would report what they observed of the red team's access, but not continue to block and terminate it.

See **Figure 1** for a timeline of the red team's activity with key points access. See the following sections for additional details, including the red team's TTPs.

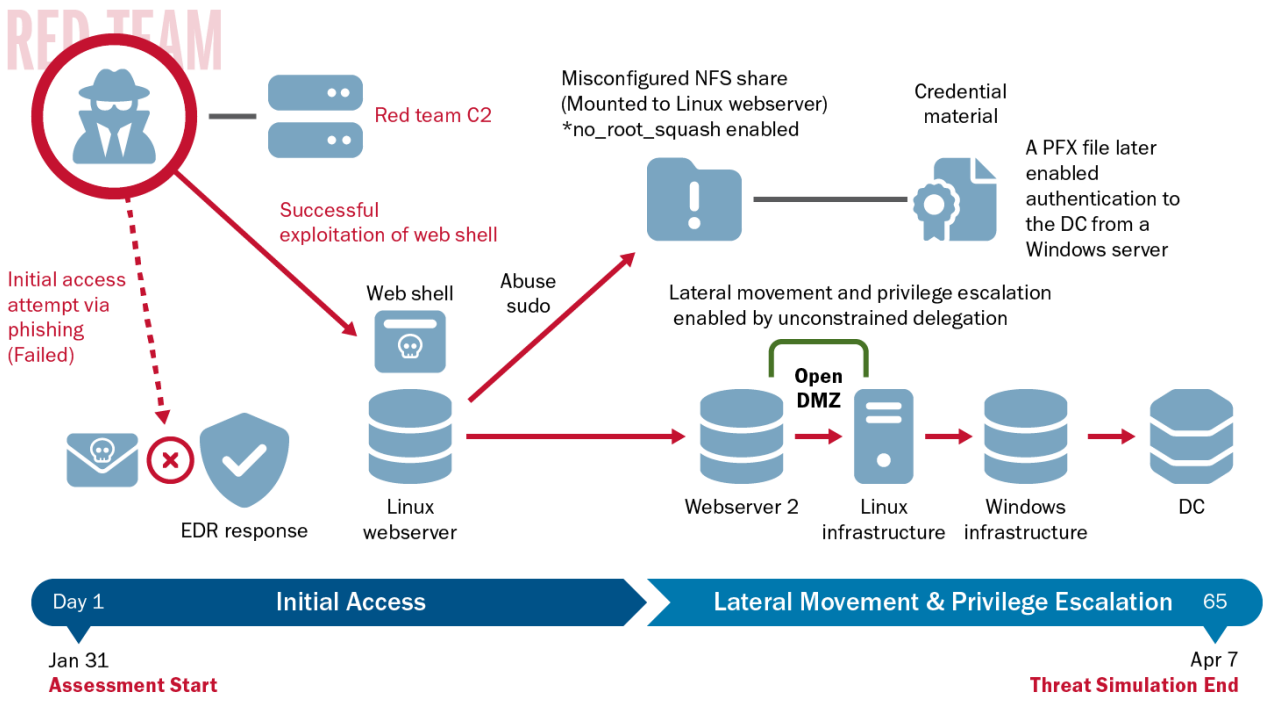

Figure 1: Timeline of Red Team Cyber Threat Activity

Initial Access

Following an unsuccessful spearphishing campaign, the red team gained initial access to the target by exploiting an internet-facing Linux web server [T1190] discovered through reconnaissance [TA0043] of the organization's external internet protocol (IP) space [T1590.005].

The red team first conducted open source research [T1593] to identify information about the organization's network, including the tools used to protect the network and potential targets for spearphishing. The red team looked for email addresses [T1589.002] and names to infer email addresses from the organization's email syntax (discovered during reconnaissance). Following this action, the red team sent tailored spearphishing emails to 13 targets [T1566.002]. Of these 13 targets, one user responded and executed two malicious payloads [T1204.002]. However, the payloads failed to bypass a previously undiscovered technical control employed by the victim organization, preventing the red team's first attempt to gain initial access.

To find an alternate pathway for initial access, the red team conducted reconnaissance with several publicly available tools, such as Shodan and Censys, to discover accessible devices and services on the internet [T1596.005]. The red team identified an old and unpatched service with a known XML External Entity (XXE) vulnerability and leveraged a public proof of concept to deploy a web shell. The associated product had an exposed endpoint—one that system administrators should typically block from the public internet—that allowed the red team to discover a preexisting web shell on the organization's Linux web server. The preexisting web shell allowed the red team to run arbitrary commands on the server [T1059] as a user (WEBUSER1). Using the web shell, the red team identified an open internal proxy server [T1016] to send outbound communications to the internet via Hypertext Transfer Protocol Secure (HTTPS). The red team then downloaded [T1105] and executed a Sliver payload that utilized this proxy to establish command and control (C2) over this host, calling back to their infrastructure [TA0011].

Note: Because the web shell and unpatched vulnerability allowed actors to easily gain initial access to the organization, the CISA red team determined this was a critical vulnerability. CISA reported both the vulnerability and the web shell to the organization in an official vulnerability notification so the organization could remediate both issues. Following this notification, the victim organization initiated threat hunting activities, detecting some of the red team's activity. The TAs determined that network defenders had previously identified and reported the vulnerability but did not remediate it. Further, the TAs found that network defenders were unaware of the web shell and believed it was likely leftover from prior VDP activity. See the **Defense Evasion and Victim Network Defense Activities** section for more information.

Linux Infrastructure Compromise

Local Privilege Escalation and Credential Access

The red team then moved laterally from the web server to the organization's internal network using valid accounts [T1078] as the DMZ was not properly segmented from the organization's internal domain.

The red team acquired credentials [TA0006] by first escalating privileges on the web server. The team discovered that WEBUSER1 had excessive `sudo` rights, allowing them to run some commands as root commands without a password. They used these elevated rights to deploy a new callback with root access [T1548.003].

With root access to the web server, the team had full access to the organization's directories and files on a NFS share with `no_root_squash` enabled. If `no_root_squash` is used, remote root users can read and change any file on the shared file system and leave a trojan horse [T1080] for other users to inadvertently execute. On Linux operating systems this option is disabled by default, yet the organization enabled it to accommodate several legacy systems. The organization's decision to enable the `no_root_squash` option allowed the red team to read all the files on the NFS share once it escalated its privileges on a single host with the NFS share mounted. This NFS share hosted the home directories of hundreds of Linux users—many of which had privileged access to one or more servers—and was auto-mounted when those users logged into Linux hosts in the environment.

The red team used its escalated privileges to search for private certificate files, Secure Shell (SSH) private keys, passwords, bash command histories [T1552.003], and other sensitive data across all user files on the NFS share [T1039]. The team initially obtained 61 private SSH keys [T1552.004] and a file containing valid cleartext domain credentials (DOMAINUSER1) that the team used to authenticate to the organization's domain [T1078.002].

Linux Command and Control

In the organization's Linux environment, the red team leveraged HTTPS connections for C2 [T1071.001]. Most of the Linux systems could not directly access the internet, but the red team circumvented this by leveraging an open internal HTTPS proxy [T1090.001] for their traffic.

Lateral Movement and Persistence

The red team's acquisition of SSH private keys generated for user and service accounts facilitated unrestricted lateral movement to other Linux hosts [T1021.004]. This acquisition included two highly privileged accounts with root access to hundreds of servers. Within one week of initial access, the team moved to multiple Linux servers and established persistence [TA0003] on four. The team used a different persistence mechanism on each Linux host, so network defenders would be less likely to discover the red team's presence on all four hosts. The team temporarily backdoored several scripts run at boot time to maintain persistence [T1037], ensuring the original versions of the scripts were re-enabled once the team successfully achieved persistence. Some of the team's techniques included modifying preexisting scripts run by the `cron` utility [T1053.003] and `ifup-post` scripts [T1037.003].

Of note, the team gained root access to an SBS-adjacent infrastructure management server that ran Ansible Tower. Access to this Ansible Tower system [T1072] provided easy access to multiple SBSs. The team discovered a root SSH private key on the host, which allowed the team to move to six SBSs across six different sensitive IP ranges. A week after the team provided screenshots of root access to the SBSs to the TAs, the TAs deconflicted the red team's access to the Ansible Tower system that network defenders discovered. The organization detected the compromise by observing abnormal usage of the root SSH private key. The root SSH private key was used to log into multiple hosts at times and for durations outside of preestablished baselines. In a real compromise, the organization would have had to shut down the server, significantly impacting business operations.

Windows Domain Controller Compromise

Approximately two weeks after gaining initial access, the red team compromised a Windows domain controller. This compromise allowed the team to move laterally to all domain-joined Windows hosts within the organization.

To first gain situational awareness about the organization's environment, the red team exfiltrated Active Directory (AD) information [TA0010] from a compromised Linux host that had network access to a Domain Controller (DC). The team queried Lightweight Directory Access Protocol (Over SSL)—(LDAPS)—to collect information about users [T1087.002], computers [T1018], groups [T1069.002], access control lists (ACL), organizational units (OU), and group policy objects (GPO) [T1615]. Unfortunately, the organization did not have detections to monitor for anomalous LDAP traffic. A non-privileged user querying LDAP from the organization's Linux domain should have alerted network defenders.

The red team observed a total of 42 hosts in AD that were not DCs, but had `Unconstrained Delegation` enabled. Hosts with `Unconstrained Delegation` enabled store the Kerberos TGTs of any user that authenticates to them. With sufficient privileges, an actor can obtain those tickets and impersonate associated users. A compromise of any of these hosts could lead to the escalation of privileges within the domain. Network defenders should work with system administrators to determine whether `Unconstrained Delegation` is necessary for their systems and limit the number of systems with `Unconstrained Delegation` unnecessarily enabled.

The red team observed insufficient network segmentation between the organization's Linux and Windows domains. This allowed for Server Message Block (SMB) and Kerberos traffic to a DC and a domain server with `Unconstrained Delegation` enabled (UDHOST). The team discovered an unprotected Personal Information Exchange (`.pfx`) file on the NFS home share that they believed was for UDHOST based on its naming convention.

Equipped with the `.pfx` file, the red team used Rubeus—an open source toolset for Kerberos interaction and abuses—to acquire a TGT and New Technology Local Area Network Manager (NTLM) hash for UDHOST from the DC. The team then used the TGT to abuse the Server-for-User-to-Self (S4U2Self) Kerberos extension to gain administrative access to UDHOST.

The red team leveraged this administrative access to upload a modified version of Rubeus in monitor mode to capture incoming tickets [T1040] on UDHOST with Rubeus' /monitor command. Next, the team ran `DFSCoerce.py` to force the domain controller to authenticate to UDHOST [T1187]. The team then downloaded the captured tickets from UDHOST.

With the DC's TGT, the team used Domain Controller Sync (DCSync) through their Linux tunnels to acquire the hash of several privileged accounts—including domain, enterprise, and server administrators—and the critical `krbtgt` account [T1003.006].

> *Gaining access to AD is not unusual for most of CISA's Red Team engagements, but it is rare to find network defenders who can secure and monitor it quickly and effectively.*

Once the team harvested the credentials needed, they moved laterally to nearly any system in the Windows domain (see **Figure 2**) through the following steps (hereafter, this combination of techniques is referred to as the "Preferred Lateral Movement Technique"):

1. The team either forged a golden ticket using the `krbtgt` hash or requested a valid TGT using the hashes they exfiltrated for a specific account before loading the ticket into their session for additional authentication.
2. The team dropped an inflated Dynamic Link Library (DLL) file associated with legitimate scheduled tasks on the organization's domain.
3. When the scheduled task executed on its own or through the red team's prompting, the DLL hijack launched a C2 implant.

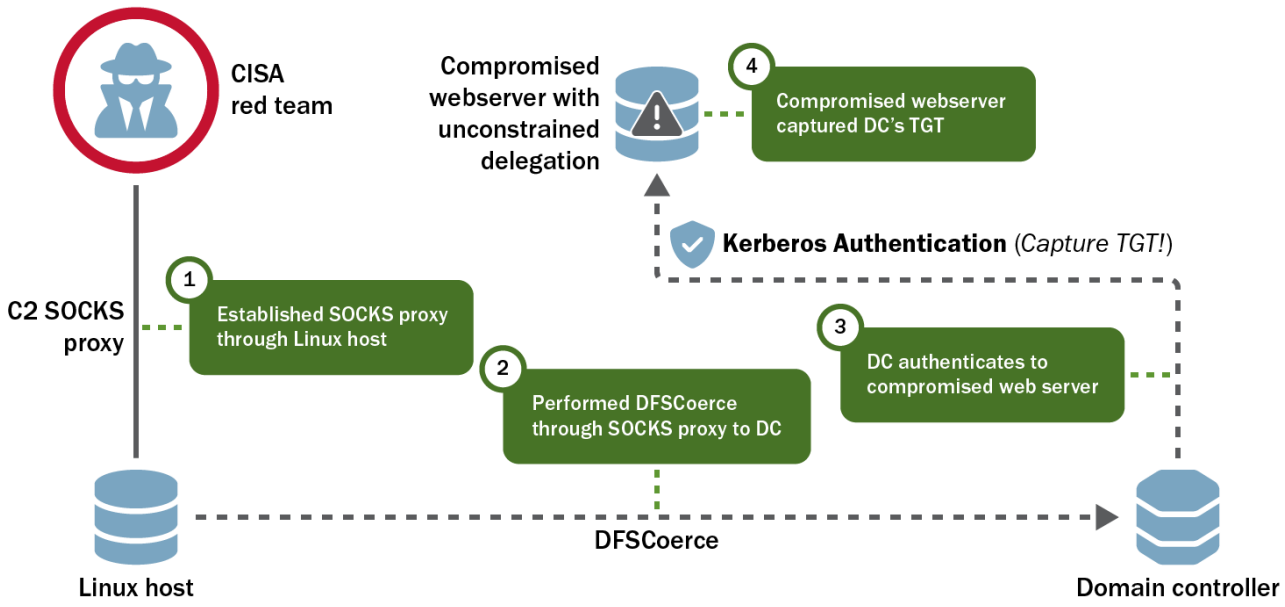

Figure 2: Movement to Domain Controller

Windows Command and Control

The red team initially established C2 on a workstation over HTTPS before connecting to servers over SMB [T1071.002] in the organization's Windows environment. To connect to certain SBSs later in its activity, the team again relied on HTTPS for C2.

Post-Exploitation Activity: Gaining Access to SBSs

After the red team gained persistent access to Linux and Windows systems across the organization's networks, the team began post-exploitation activities and attempted to access SBSs. The TAs provided a scope of the organization's Classless Inter-Domain Routing (CIDR) ranges that contained SBSs. The team gained root access to multiple Linux servers in these ranges. The TAs then instructed the red team to exploit its list of primary targets: admin workstations and network ranges that included OT networks. The

team only achieved access to the first two targets and did not find a path to the OT networks. While the team was able to affect the integrity of data derived from OT devices and applications, it was unable to find and access the organization's internal network where the OT devices resided.

To gain access to the SBSs, the team first gained access to Microsoft System Center Configuration Manager (SCCM) servers, which managed most of the domain's Windows systems. To access the SCCM servers, the team leveraged their AD data to identify administrators [T1087] of these targets. One of the users they previously acquired credentials for via DCSync was an administrator on the SCCM servers. The red team then used the Preferred Lateral Movement Technique to eventually authenticate to the SCCM servers. See **Figure 3**.

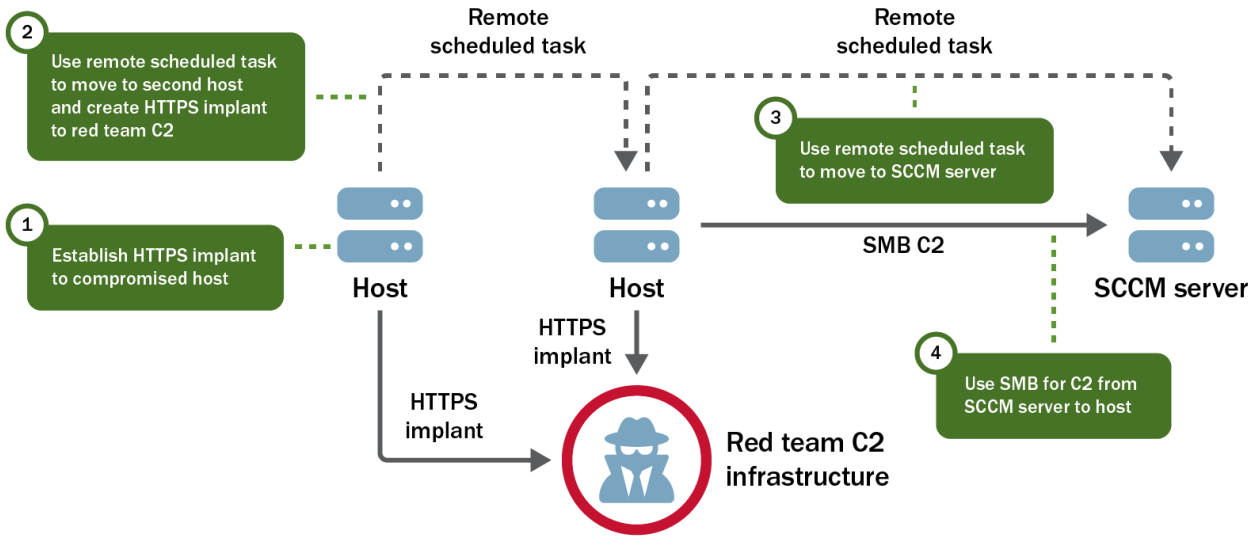

Figure 3: Attack Path to SCCM Server

Admin Workstations

The first specific set of SBS targets provided by the TAs were admin workstations. These systems are used across various sensitive networks external to, or inaccessible from, the internal network where the team already had access. Normally, authorized personnel leverage these administrator workstations to perform administrator functions. CISA's red team targeted these systems in the hopes that an authorized—but unwitting—user would move the tainted system to another network, resulting in a callback from the sensitive target network.

The red team reviewed AD data to identify these administrator systems. Through their review, the team discovered a subset of Windows workstations that could be identified with a prefix and determined a group likely to have administrative rights to the workstations.

With access to the SCCM server, the red team utilized their Preferred Lateral Movement Technique to gain access to each admin workstation target (see **Figure 4**).

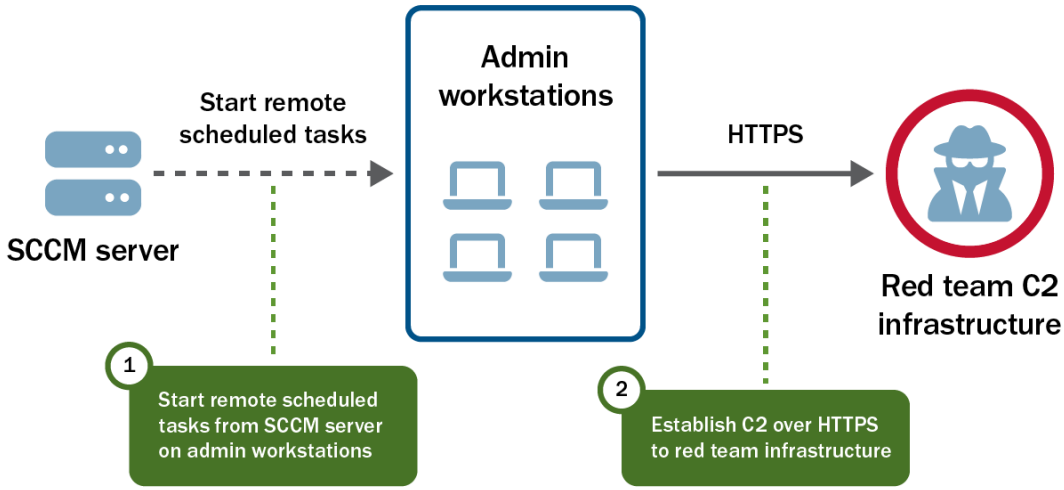

Figure 4: Attack Path from SCCM Server to Admin Workstations

The red team maintained access to these systems for several weeks, periodically checking where they were communicating from to determine if they had moved to another network. Eventually, the team lost access to these systems without a deconfliction. To the best of the red team's knowledge, these systems either did not move to new networks or, if they did, those systems no longer had the ability to communicate with red team's C2 infrastructure.

Additional Host and Other Subnets

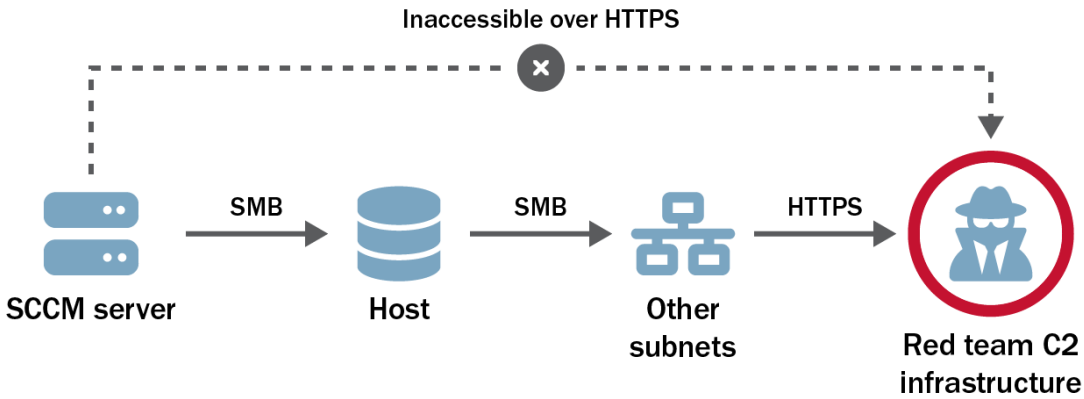

Figure 5: Attack Path from SCCM Server to Host and Other Subnets

After compromising admin workstations, the red team requested that the TAs prioritize additional systems or IP ranges. The TAs provided four CIDR ranges to target:

- A corporate DMZ that contained a mixture of systems and other subnets.

- A second subnet.
- A third subnet.
- An internal network that contained OT devices.

Access to the corporate DMZ was necessary to reach the second and third ranges, and the red team hoped that gaining access to these would facilitate access to the fourth range.

The red team followed a familiar playbook to gain access to these SBSs from another SCCM server. First, the team performed reverse DNS lookups [T1596.001] on IP addresses within the ranges the TAs provided. They then scanned SMB port `445/TCP` [T1046] from a previously compromised SCCM server to discover Windows hosts it could access on the corporate DMZ. The team discovered the server could connect to a host within the target IP range and that the system was running an outdated version of Windows Server 2012 R2. The default configuration of Windows Server 2012 R2 allows unprivileged users to query the group membership of local administrator groups. The red team discovered a user account [T1069] by querying the Windows Server 2012 R2 target that was in a database administrator group. The team leveraged its Preferred Lateral Movement Technique to authenticate to the target as that user, then repeated that technique to access a database. This database receives information from OT devices used to feed monitoring dashboards, information which factors into the organization's decision-making process [T1213].

The new host had several active connections to systems in the internal ranges of the second and third subnets. Reverse domain name system (DNS) lookup requests for these hosts failed to return any results. However, the systems were also running Windows Server 2012 R2. The red team used Windows API calls to `NetLocalGroupEnum` and `NetLocalGroupGetMembers` to query local groups [T1069.001], revealing the system names for these targets as a result. The red team performed their Preferred Lateral Movement Technique to gain access to these hosts in the second and third provided network ranges.

With access to these subnets, the red team began exploring a path to systems on a private subnet where OT devices resided but failed to locate a path to that fourth subnet.

Corporate Workstations of Critical Infrastructure Administrators and Operators

Next, the red team targeted the corporate workstations of the administrators and operators of the organization's critical infrastructure. Because the team lacked knowledge of the organization's OT devices and failed to discover a path to the private subnet where they resided, they instead tried to locate users that interacted with human machine interfaces (HMI). Access to such users could enable the team to access the HMI, which serves as a dashboard for OT.

The red team leveraged its AD data once again, combining this data with user information from SCCM to identify targets by job role and their primary workstation. Then the team targeted the desktop of a critical infrastructure administrator, the workstation of another critical infrastructure administrator, and the workstations of three critical infrastructure operators spread across two geographically disparate sites.

The AD data revealed users in a group that were administrators of all the targets. The red team then repeated their Preferred Lateral Movement Technique and identified a logged-in user connected to a "System Status and Alarm Monitoring" interface. The team discovered credentials to the interface in the user's home directory, proxied through the system, and accessed the HMI interface over HTTP. The team

did not pursue further activity involving the interface because their remaining assessment time was limited. Additionally, they did not discover a way to compromise the underlying OT devices.

Command and Control

The team used third-party owned and operated infrastructure and services [T1583] throughout its assessment, including in certain cases for command and control (C2). The tools that the red team obtained included [T1588.002]:

- Sliver, Mythic, Cobalt Strike, and other commercial C2 frameworks.
  - The team maintained multiple command and control servers hosted by several cloud vendors. They configured each server with a different domain and used the servers for communication with compromised hosts. These servers retained all assessment data.
- Two commercially available cloud-computing platforms.
  - The team used these platforms to create flexible and dynamic redirect servers to send traffic to the team's servers [T1090.002]. Redirecting servers make it difficult for defenders to attribute assessment activities to the backend team servers. The redirectors use HTTPS reverse proxies to redirect C2 traffic between the target organization's network and the team servers. The team encrypted all data in transit [T1573] and secured all data at rest through a VPN with multifactor authentication.
- Content delivery network (CDN) services.
  - This technique leverages CDNs associated with high-reputation domains, causing malicious traffic to appear directed towards a reputational domain. However, it is redirected to red team-controlled servers. This allows the team to obfuscate some of their C2 traffic.

The team used domain fronting [T1090.004] to disguise outbound traffic, diversifying communications between the domains and the persistent beacons. This technique (which also leverages CDNs) allows the beacon to appear to connect to third-party domains but instead connects to the team's redirect server.

Defense Evasion and Victim Network Defense Activities

Most of the encounters between the red team and network defenders occurred in the organization's Linux environment. The red team leveraged Linux tradecraft in an attempt to evade network defenses. In response, network defenders' threat hunting activities identified some of the team's presence in their Linux environment. To evade defenses, the red team reordered the process identifier (PID) of its executable processes to appear closer to the kernel and minimize the team's likelihood of detection. The team also modified its processes [T1055] by changing their names in memory and at execution. In addition, they used Python scripts [T1059.006] run in memory [T1620] to avoid on-disk detection. Some of the red team's Linux persistence techniques included modifying preexisting scripts run by the `cron` utility and creating backdoors through `ifup-post` scripts and `.bashrc`. Network defenders ultimately identified the team's backdoor in `.bashrc` [T1546.004].

Defenders also successfully detected anomalous activity on their Ansible Tower host and other systems in their Linux environment. The defenders actively analyzed NetFlow data, which helped them identify the red team's persistence and lateral movement. To mitigate the impact of the red team's tactics, network

defenders would have needed to shut down a critical server as part of their incident response activities. A shut down would have resulted in downtime for hundreds of systems, including SBSs.

The organization's EDR solutions largely failed to protect the organization. EDR detected only a few of the red team's payloads in the organization's Windows and Linux environments. In the instance the EDR protected the organization from the initial phishing payload, it generated an alert that network defenders neither read nor responded to. The red team excelled in bypassing EDR solutions by avoiding the use of basic "known-bad" detections the tools would capture. The team also inflated its file sizes above the upload threshold of the organization's EDR [T1027.001]. In addition, the organization completely lacked any EDR solution in a legacy environment. As such, the red team's persistence there went undetected throughout the assessment.

Network defenders failed to detect red team activity in the organization's Windows environment due to a lack of proper identity management. Specifically, network defenders failed to detect and respond to the red team's `S4U2Self`, `asktgs`, `dcsync`, and golden ticket activity. Had the organization monitored for unusual activity involving AD and Kerberos, they would have detected more red team activity.

Lastly, there were significant deficiencies in the organization's DMZ that should have detected traffic into the organization's internal network over the ports used by SMB and LDAPS. Network defenders never properly implemented an intrusion detection system (IDS) or intrusion prevention system (IPS) to monitor the DMZ at the network level. This lack of implementation resulted in a failure to detect malicious network traffic through their DMZ. Further, the red team was able to move laterally through the DMZ because there was an open internal proxy with internet access that enabled the team's C2.

Phase II: Red Team Measurable Events Activity

The red team executed 13 measurable events designed to provoke a response from the people, processes, and technology defending the organization's network. The red team did not have full visibility into the organization's responses to these events and lacked the support of a white team to help assess the organization's responses. See **Table 1** for a description of the events, the organization's actual response, and key takeaways.

Table 1: Measurable events

Measurable Event	Description	MITRE ATT&CK Technique(s)	Organization's Reported Reactions	Key Takeaways
Internal Port Scan	Launches a scan from inside the network from a previously gained workstation to enumerate ports on target workstation, server, and domain controller system(s).	Network Service Discovery [T1046]	Identified an AD enumeration tool and temporarily blocked access to one C2 domain, which resulted in the red team temporarily losing access to six SCCM servers.	The organization did not report identifying the port scan. In addition, with proper network segmentation, the scan's target hosts should not have been able to communicate.
Comprehensive Active Directory and Host Enumeration	Performs AD enumeration by querying all domain objects from the DC and enumerates trust relationships within the AD Forest, user accounts, and current session information from every domain computer (workstation and server).	Domain Trust Discovery [T1482] Account Discovery: Domain Account [T1087.002] System Owner/User Discovery [T1033] Remote System Discovery [T1018]	Identified an AD enumeration tool and temporarily blocked access to one C2 domain, which resulted in the red team temporarily losing access to six SCCM servers.	The organization did not report identifying the port scan. In addition, with proper network segmentation, the scan's target hosts should not have been able to communicate.
Data Exfiltration – 1 GB of Data	Sends a large amount (1GB) of mock sensitive information to an external host.	Exfiltration Over Alternative Protocol [T1048]	A proxy likely blocked 1GB of data exfiltration during the red team's first attempt. However, the team's second	Organizations should implement web proxies that contain data threshold restrictions. Furthermore, network defenders need to manually analyze proxy data to determine

Measurable Event	Description	MITRE ATT&CK Technique(s)	Organization's Reported Reactions	Key Takeaways
			attempt was successful when they exfiltrated the data in 100MB blocks. Hours after the data exfiltration, the organization blocked the domain used for C2 and removed access to the compromised host.	whether there is legitimate outbound traffic or potentially malicious data exfiltration.
Malicious Traffic Generation-Workstation to External Host	Establishes a session that originates from a target workstation system directly to an external host over a clear text protocol, such as HTTP.	Application Layer Protocol [T1071]	The organization blocked access to the host and the C2 domains the red team used.	The organization blocked the malicious traffic at the network level but did not appear to identify the source workstation.
Active Directory Account Lockout	Locks out several administrative AD accounts in rapid succession.	Account Access Removal [T1531]	The organization's password policy locked out the AD accounts. However, within minutes the accounts reopened, likely due to a group policy and/or an automated response.	There was no identified active response from the organization. Organizations should monitor AD account activity in Windows event logs against baselines to detect anomalous and potentially malicious activity.

Measurable Event	Description	MITRE ATT&CK Technique(s)	Organization's Reported Reactions	Key Takeaways
Local Admin User Account Creation (workstation)	Creates a local administrator account on a target workstation system.	Create Account: Local Account [T1136.001] Account Manipulation [T1098]	An automated response removed the account from local administrator's group but did not delete it.	Despite group policy objects removing the account, there were no detections for the activity.
Local Admin User Account Creation (server)	Creates a local administrator account on a target server system.	Create Account: Local Account [T1136.001] Account Manipulation [T1098]	An automated response removed the account from local Administrator's group but did not delete it.	Despite group policy objects removing the account, there were no detections for the activity.
Active Directory Account Creation	Creates AD accounts and add them to domain admins group	Create Account: Domain Account [T1136.002] Account Manipulation [T1098]	An alert existed for this action but was disabled at the time the original event was triggered, thus it was undetected. After coordination between the TAs and red team revealed this lapse, the alert was enabled, the red team performed the action once again, and this time, TAs provided a	Detection tools are only useful when network defenders tune them appropriately and effectively monitor alerts. At first, the organization missed an opportunity to respond to a tool that should have produced a true positive alert because it was misconfigured.

Measurable Event	Description	MITRE ATT&CK Technique(s)	Organization's Reported Reactions	Key Takeaways
			screenshot of the alert from their monitoring dashboards.	
Domain Admin Lateral Movement—Workstation to Domain Controller and Workstation to Workstation	Compromises a Domain Admin account and uses it to run PSExec on multiple workstations and domain controllers.	System Services: Service Execution [T1569.002] Remote Services: SMB/Windows Admin Shares [T1021.002]	None identified.	Detect malicious use of standard tools like PSExec that malicious cyber actors may use for lateral movement by monitoring Windows logs for anomalous activity. In addition, organizations should look for abnormal communications between workstations.
Malicious Traffic Generation-Domain Controller to External Host	Establishes a session that originates from a target domain controller system directly to an external host over a clear text protocol, such as HTTP.	Application Layer Protocol [T1071]	None identified.	DCs should never connect directly to an external host over HTTP. The organization failed to detect and respond to this.
Trigger Host-Based Protection-Domain Controller	Uploads and executes a well-known (e.g., with a signature) malicious file to a target DC system to generate host-based alerts.	Ingress Tool Transfer [T1105]	Malicious file was removed by host-based endpoint protection system.	Host based detection tools can be helpful in detecting known IOCs. However, organizations should focus on detecting anomalous behavior by monitoring their networks and hosts against good baselines. The blocking of this well-known tool on a DC should trigger an urgent investigation.

Measurable Event	Description	MITRE ATT&CK Technique(s)	Organization's Reported Reactions	Key Takeaways
Ransomware Simulation	Executes simulated ransomware on multiple workstation systems to simulate a ransomware attack. **Note:** This technique does not encrypt files on the target system.	N/A	Two out of nine users reported the event to defensive staff who identified all hosts that executed the ransomware. Five users likely rebooted their systems when observing the ransomware, one logged off and on, one closed the ransomware application repeatedly and continued working, one locked their screen, and another user exited the ransomware process after two hours.	Security awareness training should provide employees effective tools on how to respond to ransomware activity.

Lessons Learned and Key Findings

The red team noted the following lessons learned relevant to all organizations generated from the security assessment of the organization's network. These findings contributed to the team's ability to gain persistent access across the organization's network. See the **Mitigations** section for recommendations on how to mitigate these findings.

Lesson Learned: Insufficient Technical Controls

The assessed organization had insufficient technical controls to prevent and detect malicious activity. The organization relied too heavily on host-based EDR solutions and did not implement sufficient network layer protections.

- **Finding #1: The organization's perimeter network was not adequately firewalled from its internal network**, which allowed the red team a path through the DMZ to internal networks. A properly configured network should block access to a path from the DMZ to other internal networks.
- **Finding #2: The organization was too reliant on its host-based tools and lacked network layer protections**, such as well-configured web proxies or intrusion prevention systems (IPS). The organization's EDR solutions also failed to catch all the red team's payloads. Below is a list of some of the higher risk activities conducted by the team that were opportunities for detection:
  - Phishing;
  - Kerberoasting;
  - Generation and use of golden tickets;
  - S4U2self abuse;
  - Anomalous LDAP traffic;
  - Anomalous NFS enumeration;
  - Unconstrained Delegation server compromise;
  - DCSync;
  - Anomalous account usage during lateral movement;
  - Anomalous outbound network traffic;
  - Anomalous outbound SSH connections to the team's cloud servers from workstations; and
  - Use of proxy servers from hosts intended to be restricted from internet access.
- **Finding #3: The organization had insufficient host monitoring in a legacy environment.** The organization had hosts with a legacy operating system without a local EDR solution, which allowed the red team to persist for several months on the hosts undetected.

Lesson Learned: Continuous Training, Support, and Resources

The organization's staff requires continuous training, support, and resources to implement secure software configurations and detect malicious activity. Staff need to continuously enhance their technical

competency, gain additional institutional knowledge of their systems, and ensure are provided sufficient resources by management to adequately protect their networks.

- **Finding #4: The organization had multiple systems configured insecurely.** This allowed the red team to compromise, maintain persistence, and further exploit those systems (i.e., access credentials, elevate privileges, and move laterally). Insecure system configurations included:
  - **Default server configurations.** The organization used default configurations for hosts with Windows Server 2012 R2, which allows unprivileged users to query membership of local administrator groups. This enabled the red team to identify several standard user accounts with administrative access.
    Note: By default, NFS shares change the root user to the `nfsnobody` user, an unprivileged user account. In this way, users with local root access are prevented from gaining root level access over the mounted NFS share. Here, the organization deviated from the secure by default configuration and implemented the `no_root_squash` option to support a few legacy systems instead. This deviation from the default allowed the red team to escalate their privileges over the domain.
  - **Hosts with `Unconstrained Delegation` enabled unnecessarily.** Hosts with `Unconstrained Delegation` enabled will store the Kerberos TGTs of all users that authenticate to that host. This affords threat actors the opportunity to steal TGTs, including the TGT for a domain controller, and use them to escalate their privileges over the domain.
  - **Insecure Account Configuration.** The organization had an account running a Linux webserver with excessive privileges. The entry for that user in the `sudoers` file—which controls user rights—contained paths with wildcards where that user had write access, allowing the team to escalate privileges.
    Note: This file should only contain specific paths to executable files that a user needs to run as another user or root, and not a wildcard. Users should not have write access over any file in the `sudoers` entry.
- **Finding #5: The red team's activities generated security alerts that network defenders did not review.** In many instances, the organization relied too heavily on known IOCs and their EDR solutions instead of conducting independent analysis of their network activity compared against baselines.
- **Finding #6: The organization lacked proper identity management.** Because network defenders did not implement a centralized identity management system in their Linux network, they had to manually query every Linux host for artifacts related to the red team's lateral movement through SSH. Defenders also failed to detect anomalous activity in their organization's Windows environment because of poor identity management.

Lesson Learned: Business Risk

The organization's leadership minimized the business risk of known attack vectors for their organization. Leadership deprioritized the treatment of a vulnerability their own cybersecurity team identified, and in their risk-based decision-making, miscalculated the potential impact and likelihood of its exploitation.

- **Finding #7: The organization used known insecure and outdated software.** The red team discovered software on one of the organization's web servers that was outdated.
  - After their operations, the red team learned the insecure and outdated software was a known security concern. The organization's security team alerted management to the risks associated this software, but management accepted the risk.
  - Next, the security team implemented a VDP program, which resulted in a participant exploiting the vulnerability for initial access. The VDP program helped the security team gain management support, and they implemented a web application firewall (WAF) as a compensating control. However, they did not adequately mitigate the vulnerability as they configured the WAF to be only in monitoring mode. The security team either did not have processes (or implement them properly) to scan, assess, and test whether they treated the vulnerability effectively.

Additional Findings

The red team noted the following additional issues relevant to the security of the organization's network that contributed to their activity.

- **Unsecured Keys and Credentials.** The organization stored many private keys that lacked password protection, allowing the red team to steal the keys and use them for authentication purposes.
  - The private key of a PFX file was not password protected, allowing the red team to use that certificate to authenticate to active directory, access UDHOST, and eventually compromise the DC. In addition, the organization did not require password protection of SSH private keys.
    Note: Without a password protected key, an actor can more easily steal the private key and use it to authenticate to a system through SSH.
  - The organization had files in a home share that contained cleartext passwords. The accounts included, among other accounts, a system administrator.
    Note: The organization appeared to store cleartext passwords in the description and user password sections of Active Directory accounts. These passwords were accessible to all domain users.
- **Email Address Verification.** The active Microsoft Office 365 configuration allows an unauthenticated external user to validate email addresses through observing error messages in the form of `HTTP 302` versus `HTTP 200` responses. This misconfiguration helps threat actors verify email addresses before sending phishing emails.

Noted Strengths

The red team noted the following technical controls or defensive measures that prevented or hampered offensive actions:

- **Network defenders detected the initial compromise and some red team movement.** After being alerted of the web shell, the organization initiated hunt activities, detected initial access, and tracked some of the red team's Phase I movements. The organization terminated much of the red team's access to the organization's internal network. Of note, once the organization's defenders discovered the red team's access, the red team spent significant time and resources continuously refortifying their access to the network.
- **Host-based EDR solutions prevented initial access by phishing.** The EDR stopped the execution of multiple payloads the red team sent to a user of the organization over a week long period. The organization leveraged two products on workstations, one that was publicly discoverable and another the red team did not learn about until gaining initial access. The product the red team was unaware of, and did not test their payload against, was responsible for stopping the execution of their payloads.
- **Strong domain password policy.** The organization's domain password policy neutralized the red team's attempts to crack hashes and spray passwords. The team was unable to crack any hashes of all 115 service accounts it targeted.
- **Effective separation of privileges.** The organization's administrative users had separate accounts for performing privileged actions versus routine activities. This makes privilege escalation more difficult for threat actors.

Mitigations

Network Defenders

CISA recommends organizations implement the recommendations in **Table 2** to mitigate the findings listed in the **Lessons Learned and Key Findings** section of this advisory. These mitigations align with the Cross-Sector Cybersecurity Performance Goals (CPGs) developed by CISA and the National Institute of Standards and Technology (NIST). The CPGs provide a minimum set of practices and protections that CISA and NIST recommend all organizations implement. CISA and NIST based the CPGs on existing cybersecurity frameworks and guidance to protect against the most common and impactful threats, tactics, techniques, and procedures. See CISA's Cross-Sector Cybersecurity Performance Goals for more information on the CPGs, including additional recommended baseline protections.

Table 2: Recommendations to Mitigate Identified Findings

Finding	Recommendation
Insufficient Network Segmentation of DMZ	▪ **Apply the principle of least privilege** to limit the exposure of systems and services in the DMZ. ▪ **Segment the DMZ** based on the sensitivity of systems and services [CPG 2.F]. ▪ **Implement firewalls, access control lists, and intrusion prevention systems.**
Insufficient Network Monitoring	▪ **Establish a security baseline of normal network traffic and tune network appliances to detect anomalous behavior.** Tune host-based products to detect anomalous binaries, lateral movement, and persistence techniques [CPG 3.A]. ○ Create alerts for Windows event log authentication codes, especially for the domain controllers. This could help detect some of the pass-the-ticket, DCSync, and other techniques described in this report. ▪ **Reduce the attack surface by limiting the use of legitimate administrative pathways and tools** such as PowerShell, PsExec, and WMI, which are often used by malicious actors. Select one tool to administer the network, enable logging, and disable the others.
Insufficient Host Monitoring in Legacy Environment	▪ **Implement an EDR solution to monitor legacy hosts** for suspicious activity and to detect breaches [CPG 3.A].

Finding	Recommendation
Insecure configurations of systems	▪ **Do not use the `no_root_squash` option.** ▪ **Remove `Unconstrained Delegation` from all servers.** If `Unconstrained Delegation` functionality is required, upgrade operating systems and applications to leverage other approaches (e.g., `Constrained Delegation`) or explore whether systems can be retired or further isolated from the enterprise. ▪ **Consider disabling or limiting NTLM and WDigest Authentication** if possible. Instead, use modern federation protocols (SAML, OIDC) or Kerberos for authentication with AES-256 bit encryption. ▪ **If NTLM must be enabled, enable Extended Protection for Authentication (EPA)** to prevent NTLM-relay attacks, and implement SMB signing to prevent certain adversary-in-the-middle and pass-the-hash attacks. See Microsoft Mitigating NTLM Relay Attacks on Active Directory Certificate Services (AD CS) and Microsoft Overview of Server Message Block signing for more information. ▪ **Adhere to the principle of least privilege.** ▪ **Ensure the `sudoers` file contains only essential commands, avoids the use of wildcards, and contains password requirements** for command execution.
Lack centralized identity management and monitoring systems	▪ **From a detection standpoint, focus on identity and access management (IAM)** rather than just network traffic or static host alerts. ▪ **Examine who is accessing a resource, what is being accessed, where the request originates, and the time of activity.**
Use of known insecure and outdated software	▪ **Keep systems and software up to date**. If updates cannot be uniformly installed, update insecure configurations to meet updated standards.

Finding	Recommendation
Insecure Keys and Credentials	▪ **Implement a password protection policy for all certificates that contain private keys** that ensures every certificate is encrypted with a strong password. Ensure all certificates are stored in a secure location [CPG 2.L]. ▪ **Regularly audit network shares** to identify files that contain passwords accessible to multiple users [CPG 2.L]. ▪ **Provide training on the proper use of password management tools.** ▪ **Implement a policy that prohibits storing passwords in plaintext**, and regularly review and audit Active Directory for plain text passwords [CPG 2.L]. ▪ If system administrators must store passwords in active directory, **restrict access to only users who require them**.

Additionally, CISA recommends organizations implement the mitigations below to improve their cybersecurity posture:

- **Provide users with regular training and exercises**, specifically related to phishing emails. Phishing accounts for majority of initial access intrusion events.
- **Enforce phishing-resistant MFA to the greatest extent possible**.
- **Reduce the risk of credential compromise** via the following:
  - **Place domain admin accounts in the protected users group** to prevent caching of password hashes locally; this also forces Kerberos AES authentication as opposed to weaker RC4 or NTLM authentication protocols.
  - **Upgrade to Windows Server 2019 or greater and Windows 10 or greater**. These versions have security features not included in older operating systems.

As a long-term effort, CISA recommends organizations **prioritize implementing a more modern, Zero Trust network architecture** that:

- Leverages secure cloud services for key enterprise security capabilities (e.g., identity and access management, endpoint detection and response, and policy enforcement).
- Upgrades applications and infrastructure to leverage modern identity management and network access practices.
- Centralizes and streamlines access to cybersecurity data to drive analytics for identifying and managing cybersecurity risks.
- Invests in technology and personnel to achieve these goals.

Software Manufacturers

The above mitigations apply to critical infrastructure organizations with on-premises or hybrid environments. Recognizing that insecure software is the root cause of many of these flaws and

responsibility should not fall on the end user, CISA urges software manufacturers to implement the following:

- **Embed security into product architecture throughout the entire software development lifecycle** (SDLC).
- **Eliminate default passwords**. Do not provide software with default passwords. To eliminate default passwords, require administrators to set a strong password [CPG 2.B] during installation and configuration.
- **Design products so that the compromise of a single security control does not result in compromise of the entire system**. For example, narrowly provision user privileges by default and employ ACLs to reduce the impact of a compromised account. This will make it more difficult for a malicious cyber actor to escalate privileges and move laterally.
- **Mandate MFA**, ideally phishing-resistant MFA, for privileged users and make MFA a default, rather than opt-in, feature.
- **Reduce hardening guide size**, with a focus on systems being secure by default. In this scenario, the red team noticed default Windows Server 2012 configurations that allowed them to enumerate privileged accounts.

  Important: Manufacturers need to implement routine nudges that are built into the product rather than relying on administrators to have the time, expertise, and awareness to interpret hardening guides.

These mitigations align with principles provided in the joint guide Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software. CISA urges software manufacturers to take ownership of improving security outcomes of their customers by applying these and other secure by design practices. By adhering to secure by design principles, software manufacturers can make their product lines secure out of the box without requiring customers to spend additional resources making configuration changes, purchasing security software and logs, monitoring, and making routine updates.

For more information on secure by design, see CISA's Secure by Design webpage. For more information on common misconfigurations and guidance on reducing their prevalence, see the joint advisory NSA and CISA Red and Blue Teams Share Top Ten Cybersecurity Misconfigurations.

Validate Security Controls

In addition to applying mitigations, CISA recommends exercising, testing, and validating your organization's security program against the threat behaviors mapped to the MITRE ATT&CK for Enterprise framework in this advisory. CISA recommends testing your existing security controls inventory to assess how they perform against the ATT&CK techniques described in this advisory.

To get started:

1. Select an ATT&CK technique described in this advisory (see **Table 3** to **Table 16**).
2. Align your security technologies against the technique.
3. Test your technologies against the technique.

4. Analyze your detection and prevention technologies' performance.
5. Repeat the process for all security technologies to obtain a set of comprehensive performance data.
6. Tune your security program, including people, processes, and technologies, based on the data generated by this process.

CISA recommends continually testing your security program, at scale, in a production environment to ensure optimal performance against the MITRE ATT&CK techniques identified in this advisory.

Resources

- See CISA's RedEye tool on CISA's GitHub page. RedEye is an interactive open source analytic tool used to visualize and report red team command and control activities. See CISA's RedEye tool overview video for more information.
- See CISA's Phishing Guidance.
- See CISA's Secure by Design page to learn more about secure by design principles.

Appendix: MITRE ATT&CK Tactics and Techniques

See **Table 3** to **Table 16** for all referenced red team tactics and techniques in this advisory. **Note:** Unless noted, activity took place during Phase I. For assistance with mapping malicious cyber activity to the MITRE ATT&CK framework, see CISA and MITRE ATT&CK's Best Practices for MITRE ATT&CK Mapping and CISA's Decider Tool.

Table 3: Reconnaissance

Technique Title	ID	Use
Gather Victim Network Information	T1590	The team conducted open source research on the target organization to gain information about its network.
Gather Victim Network Information: Network Security Appliances	T1590.006	The team conducted open source research on the target organization to gain information about its defensive tools.
Gather Victim Identity Information: Employee Names	T1589.003	The team conducted open source research on the target organization to gain information about its employees.
Active Scanning	T1595	The team conducted external reconnaissance of the organization's network.
Gather Victim Network Information: IP Addresses	T1590.005	The team conducted reconnaissance of the organization's external IP space.
Search Open Websites/Domains	T1593	The team conducted open source research to identify information about the organization's network.
Gather Victim Identity Information: Email Addresses	T1589.002	The team looked for email addresses and names to infer email addresses from the organization's email syntax.
Search Open Technical Databases: Scan Databases	T1596.005	The team conducted reconnaissance with several publicly available tools, such as Shodan and Censys, to discover accessible devices and services on the internet.
Search Open Technical Databases: DNS/Passive DNS	T1596.001	The team performed reverse DNS lookups on IP addresses within the ranges the TAs provided.

Table 4: Resource Development

Technique Title	ID	Use
Acquire Infrastructure	T1583	The team used third-party owned and operated infrastructure and services throughout its assessment.
Obtain Capabilities: Tool	T1588.002	The team obtained tools (i.e., Sliver, Mythic, Cobalt Strike, and other commercial C2 frameworks).

Table 5: Initial Access

Technique Title	ID	Use
Phishing	T1566	The team designed spearphishing campaigns tailored to employees of the organization most likely to communicate with external parties.
Exploit Public-Facing Application	T1190	The team gained initial access to the target by exploiting an internet-facing Linux web server.
Phishing: Spearphishing Link	T1566.002	The team sent tailored spearphishing emails to 13 targets.

Table 6: Execution

Technique Title	ID	Use
User Execution	T1204	The team's phishing attempts were ultimately unsuccessful; targets ran the payloads, but their execution did not result in the red team gaining access into the network.
User Execution: Malicious File	T1204.002	One user responded and executed two malicious payloads.
Command and Scripting Interpreter	T1059	The preexisting web shell allowed the team to run arbitrary commands on the server.
Command and Scripting Interpreter: Python	T1059.006	The team used python scripts.
System Services: Service Execution	T1569.002	The team compromised a Domain Admin account and used it to run PSExec on multiple workstations and a domain controller.

Technique Title	ID	Use
Remote Services: SMB/Windows Admin Shares	T1021.002	The team established a session that originated from a target.

Table 7: Persistence

Technique Title	ID	Use
Server Software Component: Web Shell	T1505.003	After the failed spearphishing campaigns, the red team continued external reconnaissance of the network and discovered a web shell left from a previous VDP program.
Boot or Logon Initialization Scripts	T1037	The team backdoored several scripts run at boot time for persistence.
Scheduled Task/Job: Cron	T1053.003	Some of the team's techniques included modifying preexisting scripts run by the `cron` utility and `ifup-post` scripts.
Boot or Logon Initialization Scripts: Network Logon Script	T1037.003	The team modified preexisting scripts run by the `cron` utility and `ifup-post` scripts.
Event Triggered Execution: Unix Shell Configuration Modification	T1546.004	The team used a backdoor in `.bashrc`.
Create Account: Local Account	T1136.001	During Phase II, the team created a local administrator account on a target server system.
Account Manipulation	T1098	During Phase II, the team created a local administrator account on a target server system.
Create Account: Domain Account	T1136.002	The team created AD accounts and added them to domain admins group.

Table 8: Privilege Escalation

Technique Title	ID	Use
Valid Accounts	T1078	The team moved laterally from the web server to the organization's internal network using valid accounts.

Technique Title	ID	Use
Abuse Elevation Control Mechanism: Sudo and Sudo Caching	T1548.003	The team discovered that WEBUSER1 had excessive `sudo` rights, allowing them to run some commands as root without a password.

Table 9: Defense Evasion

Technique Title	ID	Use
Process Injection	T1055	The team modified its processes by changing their names in memory and at execution.
Reflective Code Loading	T1620	The team used Python scripts run in memory to avoid on-disk detection.
Obfuscated Files or Information: Binary Padding	T1027.001	The team inflated its file sizes above the upload threshold of the organization's EDR.

Table 10: Credential Access

Technique Title	ID	Use
Unsecured Credentials: Credentials In Files	T1552.001	The team discovered credential material on a misconfigured Network File System.
Steal or Forge Authentication Certificates	T1649	The team used a certificate for client authentication discovered on the NFS share to compromise a system configured for `Unconstrained Delegation`.
Steal or Forge Kerberos Tickets: Golden Ticket	T1558.001	The team acquired a ticket granting ticket for a domain controller.
Unsecured Credentials: Bash History	T1552.003	The team used its escalated privileges to search bash command histories.
Data from Network Shared Drive	T1039	The team used its escalated privileges to search for private certificate files, Secure Shell (SSH) private keys, passwords, bash command histories, and other sensitive data across all user files on the NFS share.
Unsecured Credentials: Private Keys	T1552.004	The team initially obtained 61 private SSH keys and a file containing valid cleartext domain credentials.

Technique Title	ID	Use
Valid Accounts: Domain Accounts	T1078.002	The team initially obtained 61 private SSH keys and a file containing valid cleartext domain credentials.
Network Sniffing	T1187	The red team leveraged this administrative access to upload a modified version of Rubeus in monitor mode to capture incoming tickets.
OS Credential Dumping: DCSync	T1003.006	The team used DCSync through Linux tunnels to acquire the hash of several privileged accounts.

Table 11: Discovery

Technique Title	ID	Use
System Network Configuration Discovery	T1016	The team leveraged the web shell to identify an open internal proxy server.
Account Discovery	T1087	The team leveraged their AD data to identify administrators of the SCCM servers.
Account Discovery: Domain Account	T1087.002	The team queried LDAPS to collect information about users, computers, groups, access control lists (ACL), organizational units (OU), and group policy objects (GPO). During Phase II, the team performed AD enumeration by querying all domain objects from the DC, as well as enumerating trust relationships within the AD Forest, user accounts, and current session information from every domain computer.
Remote System Discovery	T1018	The team queried LDAPS to collect information about users, computers, groups, access control lists (ACL), organizational units (OU), and group policy objects (GPO). During Phase II, the team performed AD enumeration by querying all domain objects from the DC as well as enumerating trust relationships within the AD Forest, user accounts, and current session information from every domain computer.

Technique Title	ID	Use
Permission Groups Discovery: Domain Groups	T1069.002	The team queried LDAPS to collect information about users, computers, groups, access control lists (ACL), organizational units (OU), and group policy objects (GPO).
Group Policy Discovery	T1615	The team queried LDAPS to collect information about users, computers, groups, access control lists (ACL), organizational units (OU), and group policy objects (GPO).
Network Service Discovery	T1046	The team scanned SMB port `445/TCP`. During Phase II, the team launched a scan from inside the network from a previously gained workstation.
Permission Groups Discovery	T1069	The team discovered a user account through querying the Windows Server 2012 R2 target.
Permission Groups Discovery: Local Groups	T1069.001	The team used Windows API calls to `NetLocalGroupEnum` and `NetLocalGroupGetMembers` to query local groups.
Domain Trust Discovery	T1482	During Phase II, the team enumerated trust relationships within the AD Forest.
System Owner/User Discovery	T1033	During Phase II, the team performed AD enumeration by querying all domain objects from the DC, as well as enumerating trust relationships within the AD Forest, user accounts, and current session information from every domain computer.

Table 12: Lateral Movement

Technique Title	ID	Use
Taint Shared Content	T1080	Since `no_root_squash` was used, the team could read and change any file on the shared file system and leave trojanized applications.

Technique Title	ID	Use
Remote Services: SSH	T1021.004	The team's acquisition of SSH private keys of user and service accounts, including two highly privileged accounts with root access to hundreds of servers, facilitated unrestricted lateral movement to other Linux hosts.
Software Deployment Tools	T1072	Access to an Ansible Tower system provided the team easy access to multiple SBSs.

Table 13: Collection

Technique Title	ID	Use
Data from Information Repositories	T1213	The team accessed a database that received information from OT devices to feed monitoring dashboards, which the organization used to make decisions.

Table 14: Command and Control

Technique Title	ID	Use
Ingress Tool Transfer	T1105	The team then downloaded and executed a Sliver payload that utilized this proxy to establish command and control. During Phase II, the team uploaded and executed a well-known malicious file to a target DC system to generate host-based alerts.
Application Layer Protocol: Web Protocols	T1071.001	In the organization's Linux environment, the red team leveraged HTTPS connections for C2.
Proxy: Internal Proxy	T1090.001	The team leveraged an open internal HTTPS proxy for their traffic.
Application Layer Protocol: File Transfer Protocols	T1071.002	The team connected to servers over SMB.
Proxy: External Proxy	T1090.002	The team used cloud platforms to create flexible and dynamic redirect servers to send traffic to the team's servers.

Technique Title	ID	Use
Encrypted Channel	T1573	The team encrypted all data in transit and secured all data at rest through a VPN with multifactor authentication.
Proxy: Domain Fronting	T1090.004	The team used domain fronting to disguise outbound traffic.
Application Layer Protocol	T1071	During Phase II, the team established a session that originated from a target Workstation system directly to an external host over a clear text protocol, such as HTTP.

Table 15: Exfiltration

Technique Title	ID	Use
Exfiltration Over Alternative Protocol	T1048	During Phase II, the team sent a large amount of mock sensitive information to an external host.

Table 16: Impact

Technique Title	ID	Use
Account Access Removal	T1531	The team locked out several administrative AD accounts in rapid succession.