Like PuTTY in Admin's Hands

By Jeff Kieschnick

Published: 2025-08-23 · Archived: 2026-04-05 17:42:41 UTC

Co-author: special thanks to Nikki Stanziale for their invaluable contributions to the research, insights, and development of this blog. While not listed as a primary author, their expertise and collaboration were instrumental in shaping the final content.

Executive Summary

Cybersecurity experts often say that humans are the weakest and most easily exploited attack vector. This is usually in reference to the average end-user, and neglects to mention that administrators and highly privileged users can also fall victim to threats.

As threat actors continue to evolve their methods for initial access and compromise, it is a reminder that we are all fallible regardless of organizational role or security expertise. This blog underlines the importance of following best security practices throughout all levels of the organization without exemption.

Recently, the LevelBlue Managed Detection and Response (MDR) Security Operations Center (SOC) team handled several incidents related to compromise stemming from privileged user activity through malvertising, masquerading as the legitimate SSH tool PuTTY.

Investigation

A SentinelOne alert for high-risk indicator detection was received by the LevelBlue SOC within USM Anywhere, LevelBlue's Open XDR platform. Initial observations of alarm artifacts displayed a download of file 'PuTTY.exe' on an endpoint. The SentinelOne threat information indicated the file was signed by 'NEW VISION MARKETING LLC' which raised the first red flag, as this does not align with expectations for legitimate PuTTY. Behavioral indicators detected by SentinelOne included potential Kerberoasting, suspicious PowerShell execution, and persistence established via scheduled task.

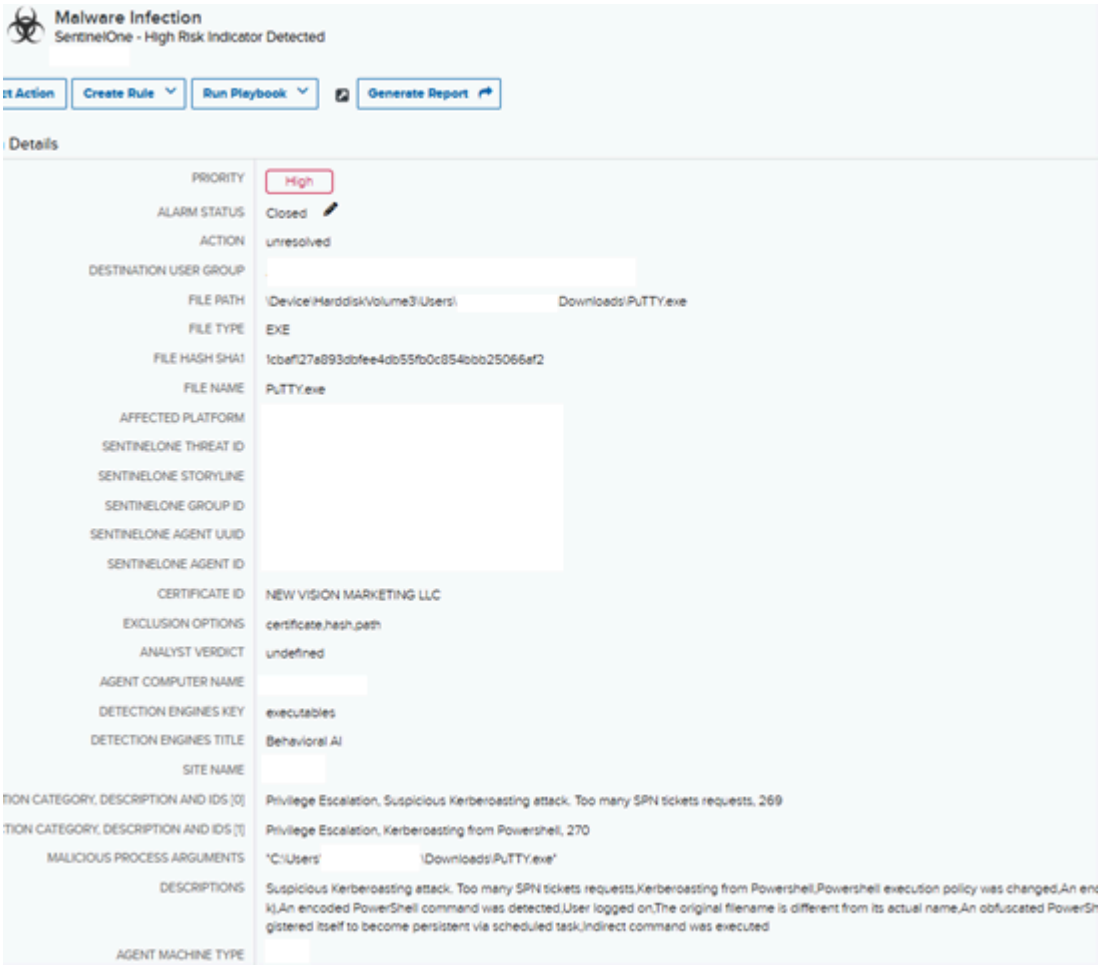

Figure 1: Screenshot of initial SentinelOne alarm received in USM displaying high-risk indicators

We began reviewing associated storyline activity within SentinelOne which raised additional red flags:

- Traffic from PuTTY.exe to two malicious IP addresses, as confirmed in VirusTotal.

- Creation of two suspicious Dynamic Link Libraries (DLLs) in the user's %appdata% and %temp% directories.

- Establishment of persistence via scheduled task that executed one of the DLLs via "rundll32.exe DllRegisterServer".

- Evidence of hands-on-keyboard (HOK) activity and Kerberoasting.

Expanded Investigation

We contacted the customer and established that this activity was anomalous and likely malicious. We immediately took action to remediate by disconnecting the affected asset from the network via SentinelOne and advising the customer to disable the user account. We used SentinelOne's Storyline feature to gain a more complete picture of what had occurred. Once downloaded, the fake PuTTY executable created a scheduled task named 'Security Updater' which was scheduled to run at three-minute intervals and executed malicious DLL 'twain_96.dll' via rundll32.exe.

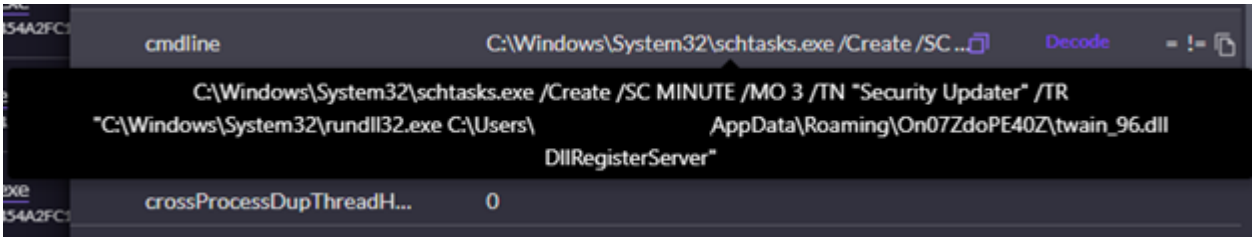

Figure 2: Scheduled task creation 'Security Updater' and parameters

The second DLL named ‘green.dll’ was dropped into the user’s %temp% folder by ‘twain_96.dll’. This DLL was recorded in a single connection event to port 443 of 144.217.206[.]26 and appeared to provide the threat actor with hands on keyboard access. This is consistent with VirusTotal results for the file hashes of ‘green.dll’ and ‘twain_96.dll’, which are reporting these files as Broomstick/Oyster malware. Broomstick/Oyster is known to provide threat actors remote command execution via cmd.exe, establish persistence via scheduled tasks that use rundll32.exe, and utilize hardcoded C2 servers – all of which were observed in this incident. The process tree seen in figure 3 shows cmd.exe spawning from the execution of rundll32.exe with “green.dll” and executing multiple discovery and recon commands via cmd.exe. The following known ransomware operator discovery TTPs were observed:

• nltest /trusted_domains

• net group “domain admins” /domain

• nltest /dclist:

The final action recorded in activity from the threat actor was the execution of an inline PowerShell script used for Kerberoasting.

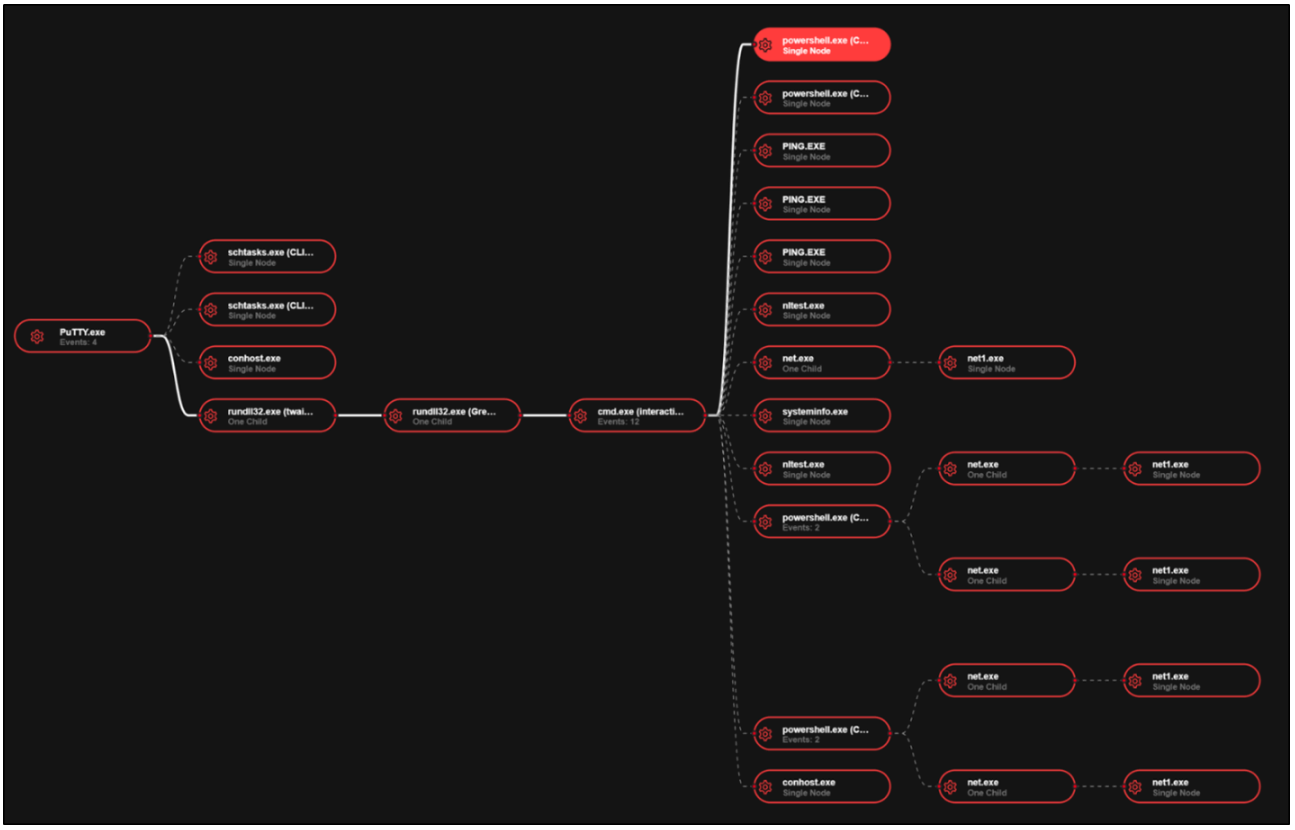

Figure 3: SentinelOne process tree from incident

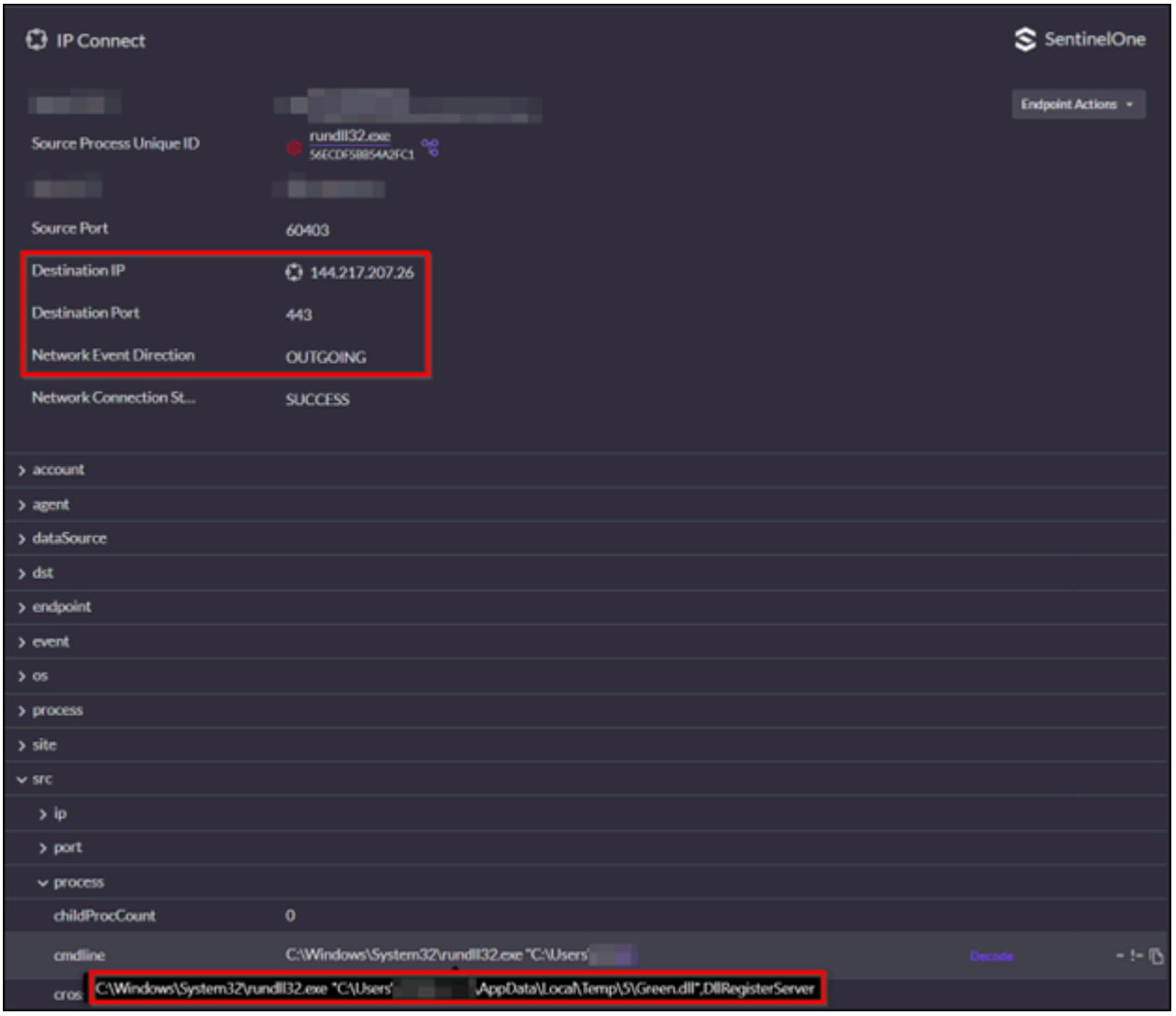

Figure 4: Green.dll connection event in S1

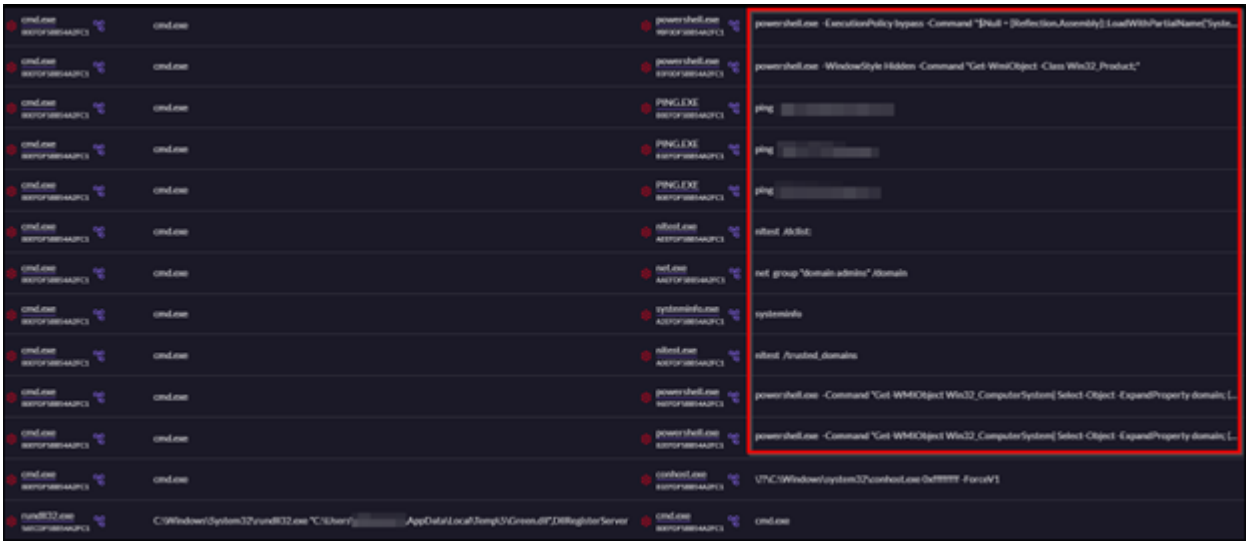

Figure 5: Hands on Keyboard activity by threat actor

Kerberoasting Script Analysis:

Kerberoasting is a well-known attack technique used to attack Active Directory service accounts by exploiting the Kerberos authentication protocol. In a Kerberoasting attack, a threat actor who has access to a valid domain user account requests Kerberos service tickets for accounts that have a SPN (Service Principal Name) defined. This is possible because Active Directory allows any domain user to request a Kerberos service ticket for accounts that have a defined SPN. The Kerberos service ticket received is encrypted with a key derived from the service account's password.

An attacker can then extract the ticket for offline cracking and utilize a tool such as Hashcat to obtain the service account's plaintext password. Active Directory environments that still allow weak RC4-HMAC encryption and are not enforcing AES encryption for Kerberos on SPNs are most vulnerable to Kerberoasting attacks. Kerberoasting is an attractive attack technique as service accounts are frequently granted privileged access in AD environments and often have weak passwords set. A successful Kerberoasting attack can allow a threat actor to escalate privilege to a valid account that can then be used for lateral movement in an environment.

There are many well-known tools that can facilitate a Kerberoasting attack, including Rubeus, Impacket's GetUserSPNs.py, and PowerSploit's Invoke-Kerberoast. The Kerberoasting script used in this incident, depicted in figure 6 below, contains components from PowerSploit's Invoke-Kerberoast, but is streamlined and operates entirely in memory without making any writes to disk. Its usage highlights how threat actors can adapt known red-team tools and leverage LOLBINs (living-off-the-land-binaries) for malicious activity.

The PowerShell commands in the observed Kerberoasting script follow this flow:

1. Loading of the .NET assembly System.IdentityModule, which is required in order to access the .NET class System.IdentityModule.Tokens.KerberosRequestorSecurityToken used later in the script.

2. Execution of an LDAP query using the .NET class DirectoryServices.DirectorySearcher to enumerate all Active Directory user objects that have a SPN defined.

3. For each user with a SPN, a Kerberos service ticket (TGS) request is made using the .NET class System.IdentityModule.Tokens.KerberosRequestorSecurityToken. Calling this class for the ticket request results in a ticket that uses weak RC4-HMAC encryption unless AES encryption is enabled for Kerberos authentication for the SPN account.

4. In-memory extraction of the raw bytes of returned Kerberos tickets, followed by hex parsing via regex and formatting the result into a $krb5tgs$ hash that is immediately compatible for usage with the Hashcat cracking tool (Hash Mode 13100). This output is written directly to the console.

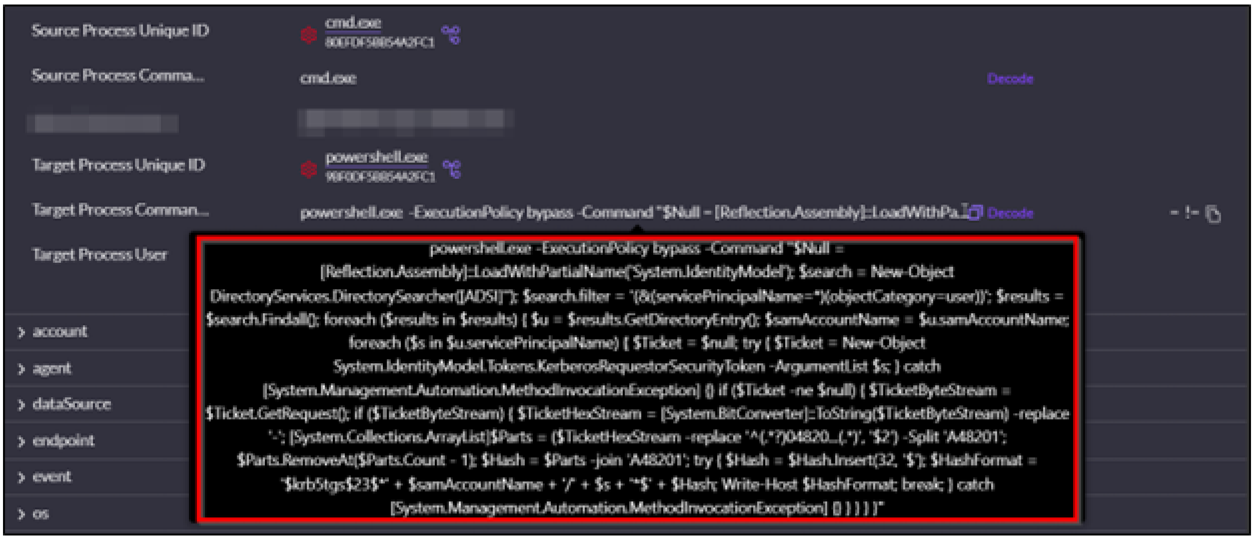

Figure 6: The Kerberoasting script executed by the threat actor

EVENT NAME	DESCRIPTION	REPORTING DEVICE RULE ID	TICKET ENCRYPTION TYPE
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC
Kerberos Service Ticket Operations	A Kerberos service ticket was requested	4769	RC4-HMAC

Figure 7: USMA events that show the RC4-HMAC encrypted Kerberos service tickets that resulted from the Kerberoasting script

Evidence of this activity was also found within the LevelBlue USM Anywhere platform in Kerberos Service Ticket events (Event ID 4769) that logged RC4-HMAC encrypted tickets. The LevelBlue MDR SOC provided our customer with a list of the SPNs recorded in the ticket requests and recommended resetting credentials for each account.

Response

While working with the customer to resolve this incident, additional members of the LevelBlue MDR SOC performed a threat hunt across our customer fleet for indicators of compromise (IOCs) observed with this

trojanized PuTTY threat.

Our team reached out to affected customers and helped them to remediate the threat prior to execution.

The LevelBlue SOC also used these indicators and observed TTPs to create new custom detection rules within SentinelOne to enhance incident detection and response times.

Additional Investigation into PuTTY Malvertising

The MDR SOC investigated further into the malvertising campaign distributing trojanized versions of the PuTTY terminal emulator. A similar campaign was active in May and June of 2024, and the recent activity appears to follow a similar playbook.

The LevelBlue team found malicious sponsored ads utilized by threat actors via Microsoft's Bing Search to deliver the trojanized PuTTY. When performing searches for "putty download" or "putty plink download", sponsored ads including those in figure 8 and 9 below were displayed in Bing Search:

Figure 8: Malicious PuTTY Ad example

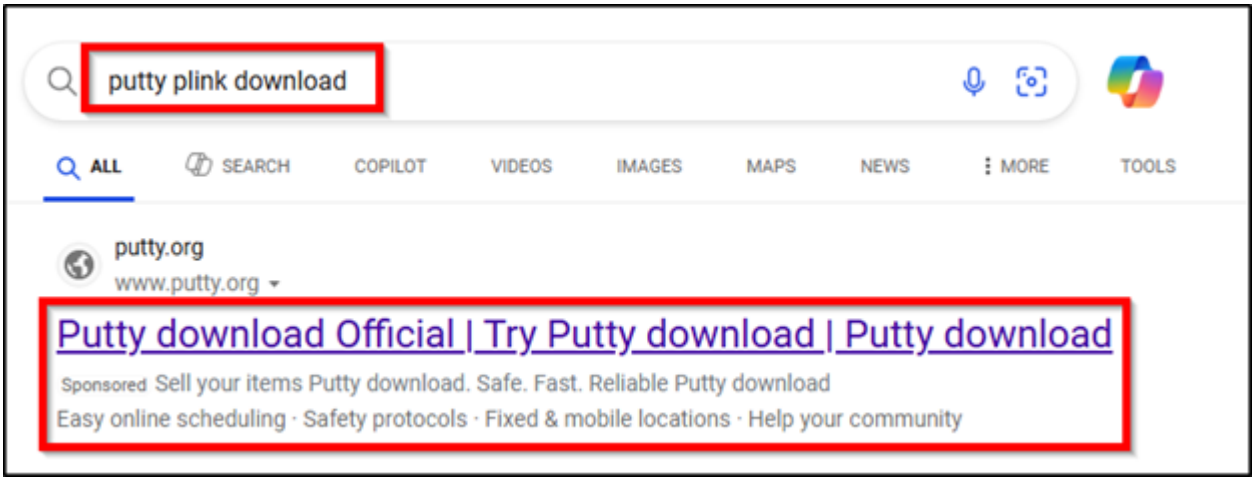

Figure 9: Malicious PuTTY Ad example

These ads were masquerading as putty[.]org, a site that is not affiliated with the official PuTTY Project but does contain download links to the official PuTTY site www.chiark.greenend.org.uk. Clicking the ad link resulted in a page setup to imitate putty[.]org but actually used a typosquatted domain such as puttyy[.]org or puttysystems[.]com. The download links on these pages were used to deliver the trojanized PuTTY. In the case of puttysystems[.]com, the LevelBlue MDR SOC observed that the domain heartlandenergy[.]ai was being used to serve the malicious payload via the 'Download PuTTY' link. A subsequent site “putty[.]network” utilized a .js script “download-script.js” that was configured to check 3 different domains (ruben.findinit[.]com, ekeitoro.siteinwp[.]com, and danielaurel[.]tv) for payload availability. The MDR SOC found that the websites for these 3 domains were all built with WordPress. WordPress vulnerabilities are commonly exploited by threat actors for drive by download and other malicious purposes and thus it seems likely the threat actor compromised these sites for payload delivery purposes.

Figure 10: Trojanized PuTTY download via puttyy[.]org

Figure 11: Trojanized PuTTY download via puttysystems[.]com

Based on the LevelBlue MDR SOC’s observations and research, they identified the following domains involved in the malvertising activity. They are all newly registered domains except for those used by the threat actors to facilitate payload delivery.

• puttyy[.]org

• puttysystems[.]com

• updaterputty[.]com

• putty[.]bet

• puttyy[.]com

• putty[.]run

• putty[.]lat

• putty[.]us[.]com

• heartlandenergy[.]ai

• putty[.]network

• ruben.findinit[.]com

• ekeitoro.siteinwp[.]com

• danielaurel[.]tv

Our team also observed that the threat actors behind this campaign consistently deployed variant forms of the malicious putty.exe payload. Multiple distinct file hashes and code-signing certificates were seen across incidents and in external research. This technique likely enhanced the campaign's effectiveness by circumventing hash-based blocklists and signature-based detection rules that relied on previously observed indicators. Additionally, a different scheduled task name was also observed in sandbox detonation of some samples – a task named "FireFox Agent INC" was observed in samples found in research after the initial incident.

The LevelBlue MDR SOC reported the malicious ad to Microsoft Advertising and received a response stating that the ad had been removed from their advertising network. While the ad did appear to have been removed, within several days our team uncovered new trojanized PuTTY payloads exhibiting the same behavior. This recurrence suggests that the threat actors are likely abusing multiple advertising platforms. It also underscores the broader issue that major advertising networks seem to lack robust verification mechanisms capable of preventing persistent abuse.

Conclusion

We recommend ensuring that all users throughout your organization undergo routine training about safe practices and device utilization. IT and Security staff should remain up-to-date on emerging threats and ensure information is appropriately disseminated to highly-privileged users.

Additionally, it is important to ensure that both in-house staff and privileged vendor accounts are using authorized and vetted administrative tools. We recommend developing a trusted repository for use within your organization and ensuring these are regularly updated and validated.

Lastly, please review the list of IOCs compiled below and add these domains to your organizational blocklist.

IOCs

Domains:

• puttyy[.]org

• puttysystems[.]com

• updaterputty[.]com

• putty[.]bet

• puttyy[.]com

• putty[.]run

• putty[.]lat

• putty[.]us[.]com

• heartlandenergy[.]ai

• putty[.]network

• ruben.findinit[.]com

• ekeitoro.siteinwp[.]com

• danielaurel[.]tv

File Hashes (SHA256):

• 0b85ad058aa224d0b66ac7fdc4f3b71145aede462068cc9708ec2cee7c5717d4

• e9f05410293f97f20d528f1a4deddc5e95049ff1b0ec9de4bf3fd7f5b8687569

• d73bcb2b67aebb19ff26a840d3380797463133c2c8f61754020794d31a9197d1

• dd995934bdab89ca6941633dea1ef6e6d9c3982af5b454ecb0a6c440032b30fb

• 03012e22602837132c4611cac749de39fb1057a8dead227594d4d4f6fb961552

• a653b4f7f76ee8e6bd9ffa816c0a14dca2d591a84ee570d4b6245079064b5794

• e02d21a83c41c15270a854c005c4b5dfb94c2ddc03bb4266aa67fc0486e5dd35

• 80c8a6ecd5619d137aa57ddf252ab5dc9044266fca87f3e90c5b7f3664c5142f

• 1112b72f47b7d09835c276c412c83d89b072b2f0fb25a0c9e2fed7cf08b55a41

• 3d22a974677164d6bd7166e521e96d07cd00c884b0aeacb5555505c6a62a1c26

• e8e9f0da26a3d6729e744a6ea566c4fd4e372ceb4b2e7fc01d08844bfc5c3abb

• eef6d4b6bdf48a605cade0b517d5a51fc4f4570e505f3d8b9b66158902dcd4af

File Signers:

• THE COMB REIVERS LIMITED

• NEW VISION MARKETING LLC

• PROFTORG LLC

• LLC Fortuna

• LLC BRAVERY

• LLC Infomed22

IPs:

• 45.86.230[.]77

• 185.208.159[.]119

• 144.217.207[.]26

• 85.239.52[.]99

• 194.213.18[.]89

URLs:

• hxxp[:]/185.208.158[.]119/api/jgfnsfnuefcnegfnehjbfncejfh

• hxxp[:]/185.208.158[.]119/api/kcehc

• hxxp[:]/45.86.230[.]77:443/reg

• hxxp[:]/45.86.230[.]77:443/login

• hxxp[:]/85.239.52[.]99/api/jgfnsfnuefcnegfnehjbfncejfh

• hxxp[:]/85.239.52[.]99/api/kcehc

• hxxp[:]/194.213.18[.]89:443/reg

• hxxp[:]/194.213.18[.]89:443/login

Scheduled Task Creations:

• Security Updater

• FireFox Agent INC

Source: https://levelblue.com/blogs/security-essentials/like-putty-in-admins-hands