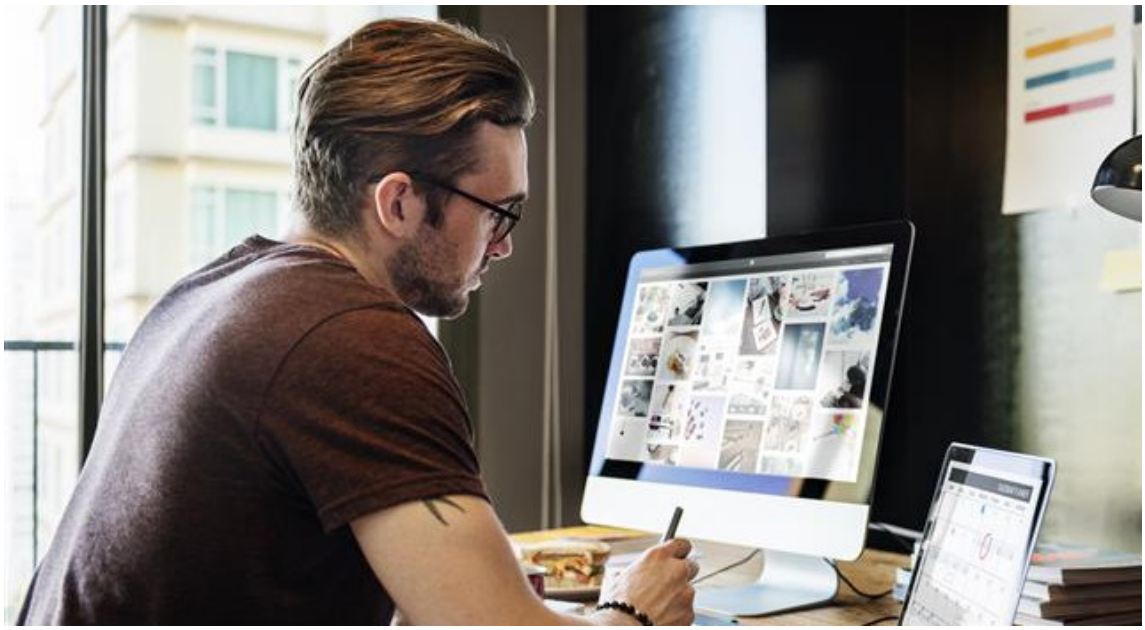
TeamTNT's Extended Credential Harvester Targets Cloud Services, Other Software

trendmicro.com/en_us/research/21/e/teamtnt-extended-credential-harvester-targets-cloud-services-other-software.html

May 18, 2021



We found new evidence that the cybercriminal group TeamTNT has extended its credential harvesting capabilities to include multiple cloud and non-cloud services.

By: David Fiser, Alfredo Oliveira May 18, 2021 Read time: (words)

Keeping secrets safe is key to keeping systems secure and preventing supply-chain attacks. Malicious actors often go after secrets in storage mechanisms and harvest credentials found in compromised systems. And credentials stored in plain text that are accessible even without user interaction are not uncommon for DevOps software and are therefore a big security risk.

Malicious actors have been known to harvest cloud service provider (CSP) credentials once they get into their victims' systems. The cybercriminal group TeamTNT, for example, is no stranger to targeting cloud containers, expanding their arsenal to steal cloud credentials, and exploring other environments and intrusive activities. In the group's latest attack routine, we found new evidence that TeamTNT has further extended its credential harvesting capabilities to target multiple cloud and non-cloud services in victims' internal networks and systems post-compromise.

Technical analysis

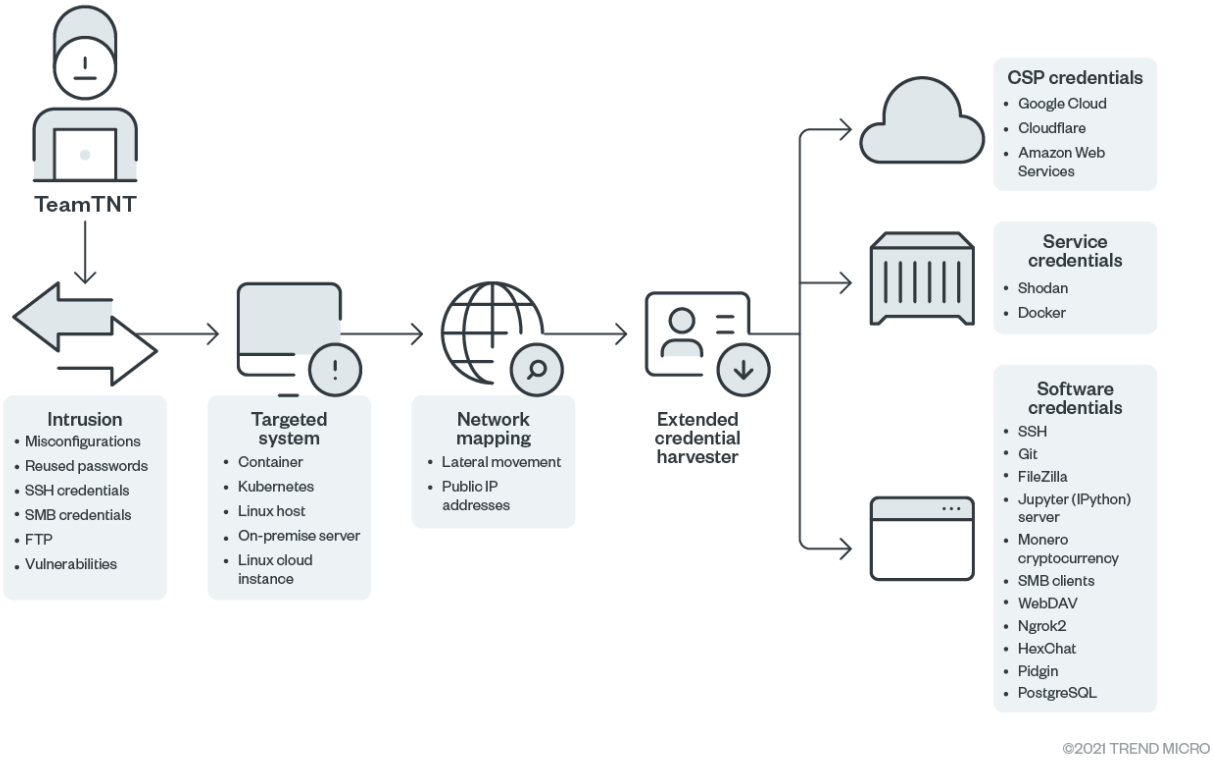


Figure 1. TeamTNT's infection chain for harvesting credentials

TeamTNT's malware is designed to harvest credentials from specific software and services. It infects Linux machines with openings such as exposed private keys and recycled passwords, and focuses on checking the infected systems for cloud-related files.

As in the group's other attacks, cloud misconfigurations and reused passwords provide easy entry to a victim system. And as before, the group harvests credentials for Secure Shell (SSH) and Server Message Block (SMB) to obtain access to other systems. Both intrusion techniques can spread their respective payloads in a wormlike manner. We found several scripts for this function (to ensure the spread of payloads in different environments), one of which we had previously documented.

With a .netrc file to automatically log in using the harvested credentials, the malware looks for app configurations and data based on a search list while going through the connected systems, and sends them to the command-and-control (C&C) server.

```
#
# SSH, AWS, Docker, s3cfg, GitHub, Shodan, gcloud,
# Ngrok, Pidgin, FileZilla, HexChat, MoneroGuiWallet,
# CloudFlared, davfs2, PostgreSQL, smbClients
#
# wget -O - [redacted]
#
clear; echo "";echo "";echo "scan for files and data of interest: ";echo "";echo ""
```

Figure 2. Scanning for configurations that the group is interested in

```
for CHECK_PATH in ${PATH_ARRAY[@]}; do
if [ "$(whoami)" = "root" ];then
if [ -f "/root/$CHECK_PATH" ];then echo -e "\e[1;33;41m FOUND:
/root/$CHECK_PATH \033[0m";fi
fi
for USER_AXX in $(ls -1 /home/); do
if [ -f "/home/$USER_AXX/$CHECK_PATH" ];then echo -e "\e[1;33;41m
FOUND: /home/$USER_AXX/$CHECK_PATH \033[0m";fi
done
```

Figure 3. Checking whether any of the configuration files of targeted services are present in the infected system

If at least one of the sought-after configuration files is present in the infected system, the extended credential harvester aggregates all the services' configuration files into two arrays. Comparing this harvester with the group's previous versions, we saw a significant increase in targets.

```
FULL_ARRAY=("/etc/passwd-s3fs" "/etc/davfs2/secrets"
"/etc/zypp/credentials.d/NCCcredentials" "/etc/cloudflared/config.yml"
"/etc/eksctl/metadata.env")

PATH_ARRAY=(".ssh/id_rsa" ".ssh/id_rsa.pub" ".ssh/known_hosts" ".ssh/config"
".ssh/authorized_keys" ".ssh/authorized_keys2" \
                        ".aws/config" ".aws/credentials" ".aws/credentials.gpg" ".docker/config.json"
".docker/ca.pem" ".s3backer_passwd" "s3proxy.conf" \
                       ".s3ql/authinfo2" ".passwd-s3fs" ".s3cfg" ".git-credentials" ".gitconfig"
".shodan/api_key" ".ngrok2/ngrok.yml" ".purple/accounts.xml" \
             ".config/filezilla/filezilla.xml"
".config/filezilla/recentservers.xml" ".config/hexchat/servlist.conf"
".config/monero-project/monero-core.conf" \
             ".boto" ".netrc" ".config/gcloud/access_tokens.db"
".config/gcloud/credentials.db" ".davfs2/secrets" ".pgpass"
".local/share/jupyter/runtime/notebook_cookie_secret" \
             ".smbclient.conf" ".smbcredentials" ".samba_credentials")
```

Figure 4. Aggregating the targeted services' configuration files into two arrays

As TeamTNT's payloads focus on unauthorized mining of Monero, it's no surprise that the malware also looks for Monero configuration files in the infected system. The malware looks for the presence of Monero wallets in all the systems the group can access.

At the end of its routine, the malware tries to delete traces of itself from the infected system. However, our analysis strongly indicates that this is not done effectively. While "history -c" clears the Bash history, some commands continue with their activities and leave traces on other parts of the system.

```
echo "";echo "";echo "done!";echo "";echo ""
history -c
sleep 3
clear
```

Figure 5. Attempting to clear traces of the routine from the infected system

Conclusion

Malicious actors actively look for legitimate users' credentials in internal networks and systems to facilitate their post-intrusion activities. If in possession of CSP credentials, they could use the cloud services paid by legitimate entities for other malicious activities. Stolen credentials for version control software such as Git also pose significant security risks, including supply chain compromise, since it is highly likely that a malicious user will also have write capability into repositories, and can therefore perform source code modifications that will go unnoticed.

In addition, credentials stored in plain text serve as a gold mine for cybercriminals, especially when used in subsequent attacks. Harvested FTP credentials, for example, could lead to old-school website hacking or credential modifications, followed by ransom demands in exchange for access or data restoration. The same goes for vulnerabilities, especially those in unpatched and otherwise unsecured internet-facing systems.

To mitigate the risks of this TeamTNT routine and other similar threats, customers are advised to use the secret vaults offered by their CSPs and to follow these best practices:

- Enforce the principle of least privilege and adopt the shared responsibility model.
- Replace default credentials with strong and secure passwords, and ensure that security settings of different systems' environments are customized to the organization's needs.
- Avoid storing credentials in plain text, and enable multifactor authentication whenever available.
- Update and patch systems regularly. Consider the security, customization of credentials, and patching of front-facing systems to ensure no gaps can be abused for malicious activities.

Trend Micro solutions

Cloud-specific security solutions such as Trend Micro Hybrid Cloud Security can help protect cloud-native systems and their various layers. Trend Micro Hybrid Cloud Security is powered by Trend Micro Cloud One™, a security services platform for cloud builders that provides automated protection for continuous-integration and continuous-delivery (CI/CD) pipelines and applications. It also helps identify and resolve security issues sooner and improve delivery time for DevOps teams. The Trend Micro Cloud One platform includes:

- Workload Security: runtime protection for workloads
- Container Security: automated container image and registry scanning
- File Storage Security: security for cloud files and object storage services
- Network Security: cloud network layer for intrusion prevention system (IPS) security
- Application Security: security for serverless functions, APIs, and applications
- Conformity: real-time security for cloud infrastructure — secure, optimize, comply

Indicator of compromise

SHA256	Detection
ed40bce040778e2227c869dac59f54c320944e19f77543954f40019e2f2b0c35	Trojan.SH.YELLOWDYE.A