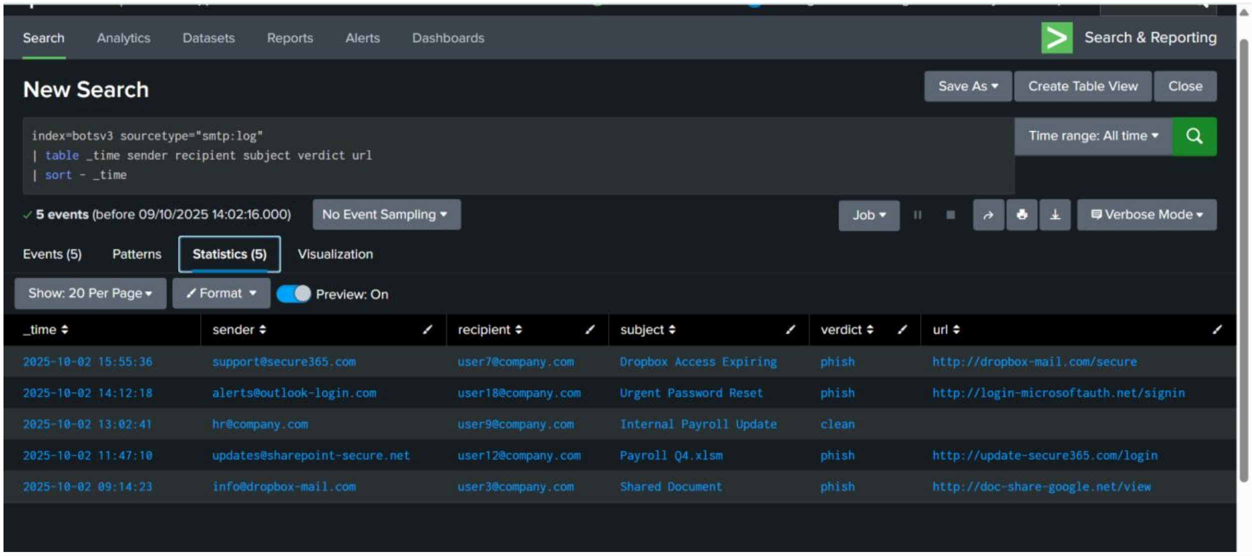
Phishing project

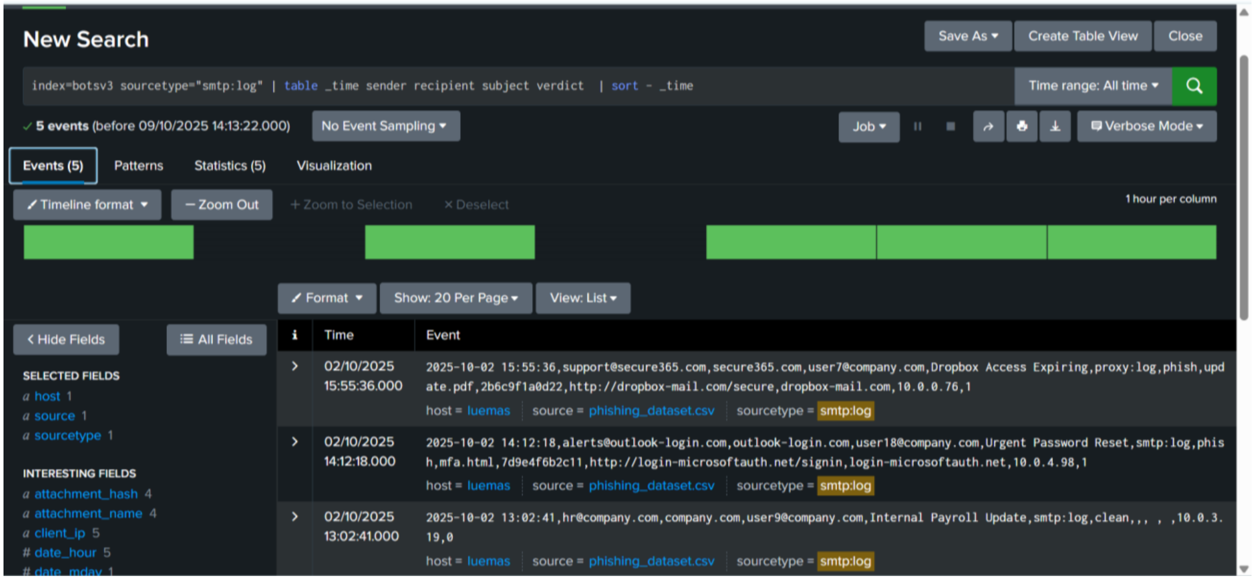
1. 1. A user reports a suspicious email with the subject 'Invoice Request' from an external domain. Using sourcetype='smtp:log', write a Splunk query to identify if multiple users received similar emails from that sender within the last 48 hours.
   Ans.



```
index=botsv3 sourcetype="smtp:log"
| table _time sender recipient subject verdict url
| sort - _time
```

✓ 5 events (before 09/10/2025 14:02:16.000)

_time	sender	recipient	subject	verdict	url
2025-10-02 15:55:36	support@secure365.com	user7@company.com	Dropbox Access Expiring	phish	http://dropbox-mail.com/secure
2025-10-02 14:12:18	alerts@outlook-login.com	user18@company.com	Urgent Password Reset	phish	http://login-microsoftauth.net/signin
2025-10-02 13:02:41	hr@company.com	user9@company.com	Internal Payroll Update	clean	
2025-10-02 11:47:10	updates@sharepoint-secure.net	user12@company.com	Payroll Q4.xlsm	phish	http://update-secure365.com/login
2025-10-02 09:14:23	info@dropbox-mail.com	user3@company.com	Shared Document	phish	http://doc-share-google.net/view



```
index=botsv3 sourcetype="smtp:log" | table _time sender recipient subject verdict  | sort - _time
```

✓ 5 events (before 09/10/2025 14:13:22.000)

SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
a attachment_hash 4
a attachment_name 4
a client_ip 5
# date_hour 5
# date_mday 1

```
i  Time                      Event
>  02/10/2025 15:55:36.000   2025-10-02 15:55:36,support@secure365.com,secure365.com,user7@company.com,Dropbox Access Expiring,proxy:log,phish,update.pdf,2b6c9f1a0d22,http://dropbox-mail.com/secure,dropbox-mail.com,10.0.0.76,1
                             host = luemas | source = phishing_dataset.csv | sourcetype = smtp:log
>  02/10/2025 14:12:18.000   2025-10-02 14:12:18,alerts@outlook-login.com,outlook-login.com,user18@company.com,Urgent Password Reset,smtp:log,phish,mfa.html,7d9e4f6b2c11,http://login-microsoftauth.net/signin,login-microsoftauth.net,10.0.4.98,1
                             host = luemas | source = phishing_dataset.csv | sourcetype = smtp:log
>  02/10/2025 13:02:41.000   2025-10-02 13:02:41,hr@company.com,company.com,user9@company.com,Internal Payroll Update,smtp:log,clean,,, , ,10.0.3.19,0
                             host = luemas | source = phishing_dataset.csv | sourcetype = smtp:log
```

Explanation / Note:

This query retrieves **email log events** from the **BOTS v3 dataset** that have the **sourcetype** **`smtp:log`**, which contains records of inbound and outbound emails processed by the mail system.
Here’s what each part does:

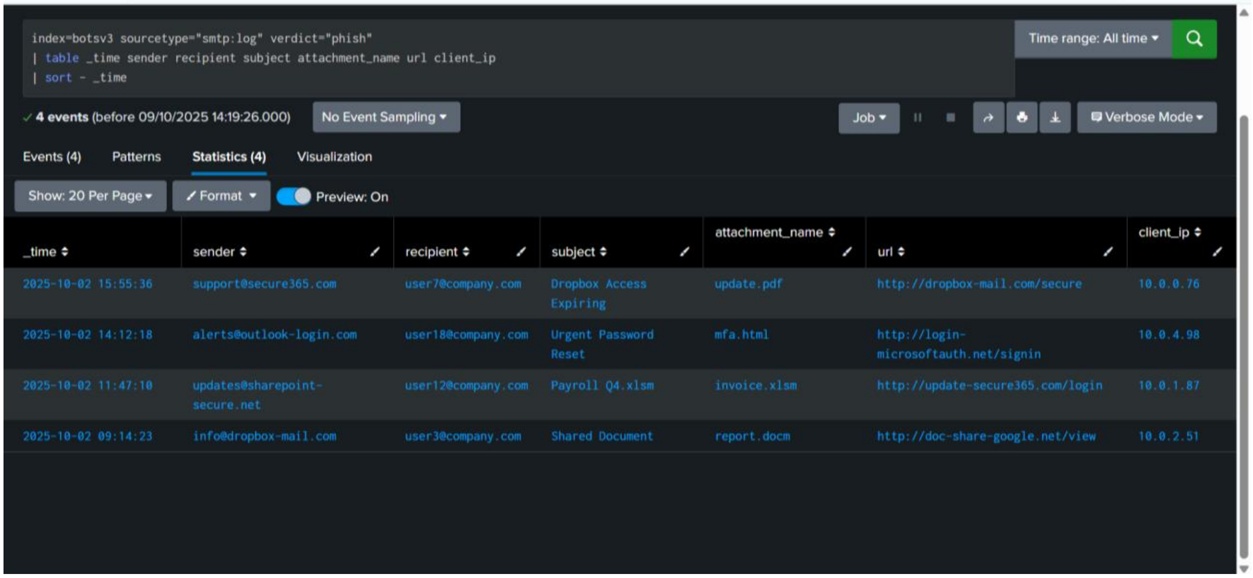
- **`index=botsv3`** → Searches within the BOTS v3 dataset where all related events are stored.

- **`sourcetype="smtp:log"`** → Filters to only include SMTP (email) logs — these represent messages sent and received by users.
- **`| table _time sender recipient subject verdict`** → Selects and formats only the key fields:
    - `_time`: Timestamp of the email event.
    - `sender`: Email address of the sender.
    - `recipient`: The target user's email address.
    - `subject`: The subject line of the email.
    - `verdict`: Classification label (e.g., "phish" or "clean").
- **`| sort - _time`** → Sorts the results in descending order of time so that the most recent emails appear first.

Purpose

This query provides a **clean, chronological view of all email traffic** in the dataset, making it easier to spot suspicious subjects, repeated senders, or emails marked as phishing.
It's often used as the **starting point for deeper investigations** or dashboard panels (like "Recent Emails Analyzed" or "Latest Phishing Alerts").

2. Your proxy logs (sourcetype='web:proxy') show several outbound requests to 'login-microsoftauth.net'. Develop an SPL query to find which users clicked that URL and from which IP addresses
   Ans.



```
index=botsv3 sourcetype="smtp:log" verdict="phish"
| table _time sender recipient subject attachment_name url client_ip
| sort - _time
```

✓ 4 events (before 09/10/2025 14:19:26.000)

_time	sender	recipient	subject	attachment_name	url	client_ip
2025-10-02 15:55:36	support@secure365.com	user7@company.com	Dropbox Access Expiring	update.pdf	http://dropbox-mail.com/secure	10.0.0.76
2025-10-02 14:12:18	alerts@outlook-login.com	user18@company.com	Urgent Password Reset	mfa.html	http://login-microsoftauth.net/signin	10.0.4.98
2025-10-02 11:47:10	updates@sharepoint-secure.net	user12@company.com	Payroll Q4.xlsm	invoice.xlsm	http://update-secure365.com/login	10.0.1.87
2025-10-02 09:14:23	info@dropbox-mail.com	user3@company.com	Shared Document	report.docm	http://doc-share-google.net/view	10.0.2.51

Explanation:

This SPL query filters and displays **only phishing emails** from the BOTS v3 dataset based on the `verdict` field. It focuses on analyzing malicious messages and their related indicators of compromise (IOCs) such as attachments, URLs, and sender details.
break it down:

- **`index=botsv3`** → Searches within the main dataset containing all event types (email, proxy, endpoint, etc.).

- **`sourcetype="smtp:log"`** → Filters to only email (SMTP) events, which record mail server logs of sent and received messages.
- **`verdict="phish"`** → Narrows down the results to events classified as phishing attempts by the email security system or dataset labeling.
- **`| table _time sender recipient subject attachment_name url client_ip`** → Formats the output to show only the most relevant fields:
  - `_time` → When the phishing email was sent or received.
  - `sender` → The (possibly spoofed) address of the attacker.
  - `recipient` → The target user's email address.
  - `subject` → The subject line — often crafted to lure the recipient (e.g., "Password Reset", "Urgent Invoice").
  - `attachment_name` → The name of any file attached (common phishing indicators: `.docm`, `.xlsm`, `.pdf`).
  - `url` → Embedded link within the message body (possible malicious redirect or credential phishing site).
  - `client_ip` → IP address that sent or relayed the email (can be traced for reputation checks).
- **`| sort - _time`** → Orders the events by most recent first for easier timeline analysis.

□ Purpose:

This query is used to:

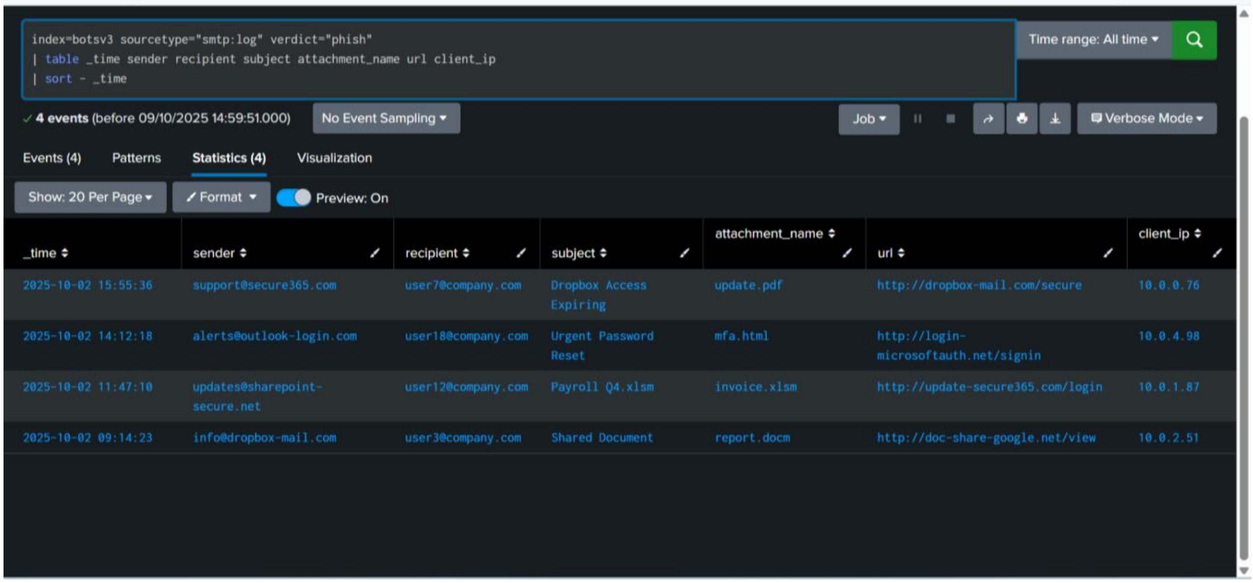
- Identify all **phishing emails** delivered to users.
- Extract indicators (malicious senders, domains, or attachments).
- Serve as a foundation for **threat-hunting dashboards** or **incident response workflows** (e.g., "Recent Phishing Emails," "Top Phishing Senders," "Common Phishing URLs").

💡 Analyst Insight:

After running this query, an analyst can:

- Look for **patterns** in senders or subjects (e.g., same domain repeatedly).
- Correlate `url` values with proxy or threat intel feeds.
- Use `client_ip` for enrichment or reverse lookup to identify malicious infrastructure.

3. . An alert from the EDR system (sourcetype='edr:telemetry') indicates PowerShell execution shortly after an email click. Correlate the EDR data with email logs to identify if the user clicked a phishing link before execution.
   Ans.



```
index=botsv3 sourcetype="smtp:log" verdict="phish"
| table _time sender recipient subject attachment_name url client_ip
| sort - _time
```

_time	sender	recipient	subject	attachment_name	url	client_ip
2025-10-02 15:55:36	support@secure365.com	user7@company.com	Dropbox Access Expiring	update.pdf	http://dropbox-mail.com/secure	10.0.0.76
2025-10-02 14:12:18	alerts@outlook-login.com	user18@company.com	Urgent Password Reset	mfa.html	http://login-microsoftauth.net/signin	10.0.4.98
2025-10-02 11:47:10	updates@sharepoint-secure.net	user12@company.com	Payroll Q4.xlsm	invoice.xlsm	http://update-secure365.com/login	10.0.1.87
2025-10-02 09:14:23	info@dropbox-mail.com	user3@company.com	Shared Document	report.docm	http://doc-share-google.net/view	10.0.2.51

Explanation / Note

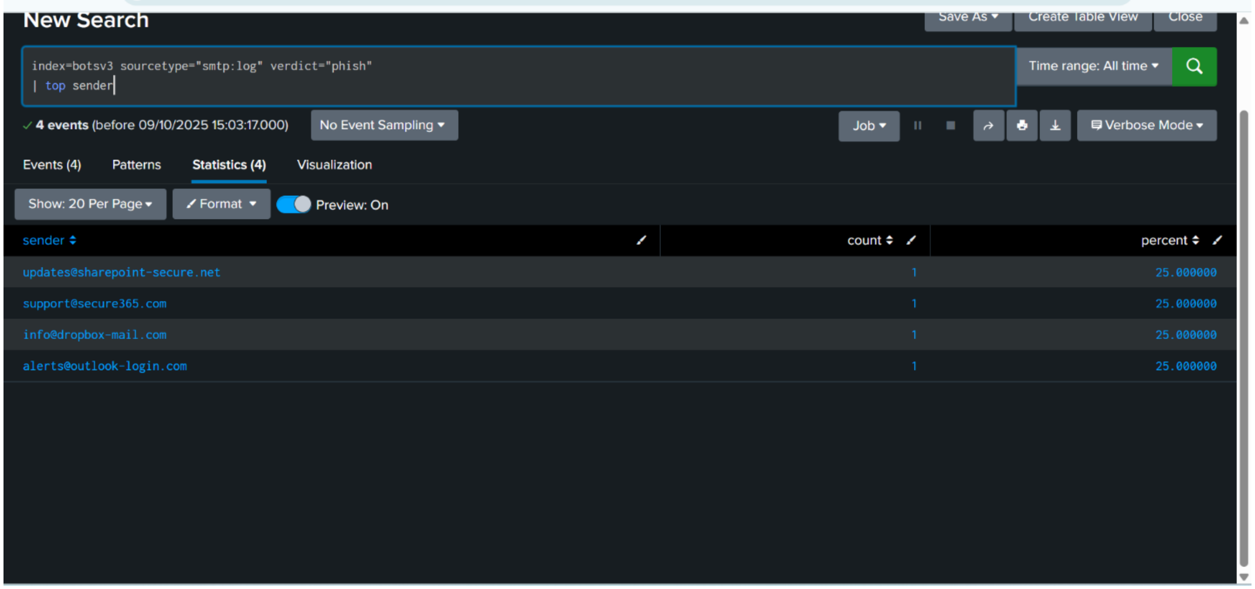
This query retrieves and displays **all phishing emails** detected in the dataset from the `smtp:log` sourcetype within the **BOTS v3 index**.

🔹 *Step-by-Step Breakdown*

- **`index=botsv3`** → Searches in the main BOTS v3 dataset.
- **`sourcetype="smtp:log"`** → Restricts results to email (SMTP) log data.
- **`verdict="phish"`** → Filters only those emails classified as phishing.
- **`| table _time sender recipient subject attachment_name url client_ip`** → Displays key fields in a readable table:
  - `_time`: When the email was received.
  - `sender`: Who sent the email (often spoofed).
  - `recipient`: Who received the email.
  - `subject`: Email subject line — helps identify lures.
  - `attachment_name`: Filename of any attachment.
  - `url`: Embedded phishing link (if any).
  - `client_ip`: IP address from which the email was sent.
- **`| sort - _time`** → Sorts results by most recent first.

Analyst Tip

Top phishing senders



```
index=botsv3 sourcetype="smtp:log" verdict="phish"
| top sender
```

✓ 4 events (before 09/10/2025 15:03:17.000)

sender	count	percent
updates@sharepoint-secure.net	1	25.000000
support@secure365.com	1	25.000000
info@dropbox-mail.com	1	25.000000
alerts@outlook-login.com	1	25.000000

Note

This query identifies the **most frequent senders** of phishing emails within the dataset, helping analysts quickly pinpoint the **most active or repeated phishing sources**.

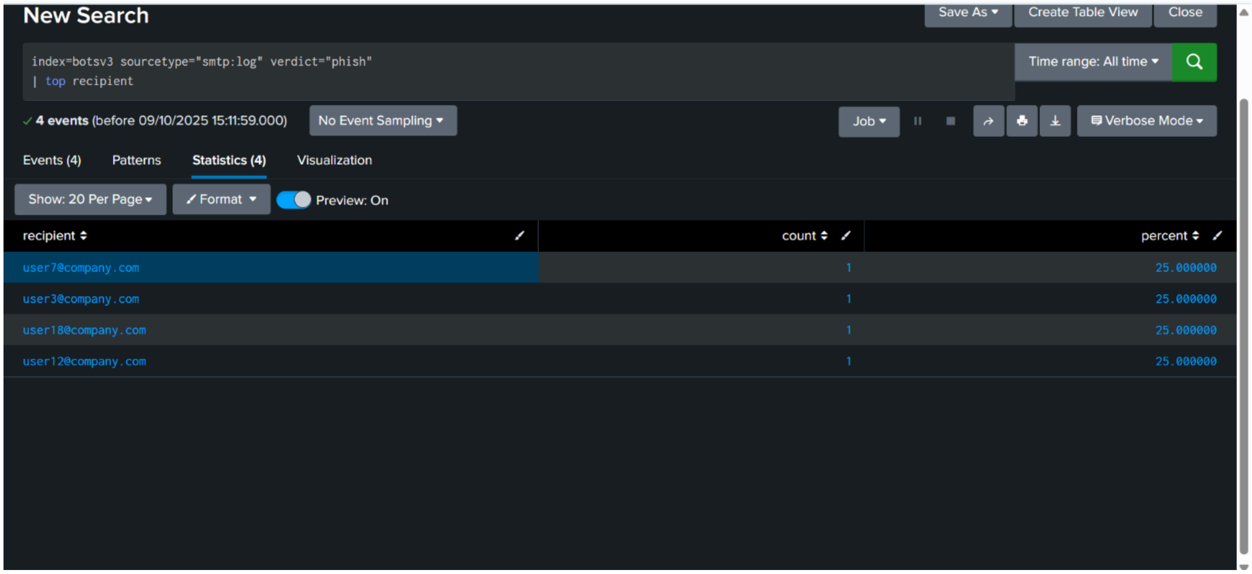
Breakdown

- **`index=botsv3`** → Searches the entire BOTS v3 dataset.
- **`sourcetype="smtp:log"`** → Focuses only on email (SMTP) logs.
- **`verdict="phish"`** → Filters to only those messages that were classified as phishing.
- **`| top sender`** → Uses Splunk's built-in `top` command to count and display the senders that appear most often in phishing events.

Purpose

This query is used to:

- Identify **the most common phishing senders** targeting users.
- Detect **repeated attack campaigns** using the same spoofed or malicious domain.
- Prioritize **blocklist or alerting actions** on the top malicious senders.
  Top targeted users



```
index=botsv3 sourcetype="smtp:log" verdict="phish"
| top recipient
```

✓ 4 events (before 09/10/2025 15:11:59.000)

recipient	count	percent
user7@company.com	1	25.000000
user3@company.com	1	25.000000
user18@company.com	1	25.000000
user12@company.com	1	25.000000

Explanation / Note

This query identifies the **users most frequently targeted by phishing emails** within the dataset. It helps analysts understand **who in the organization is being attacked most often**, which is vital for prioritizing awareness training and security controls.

Breakdown

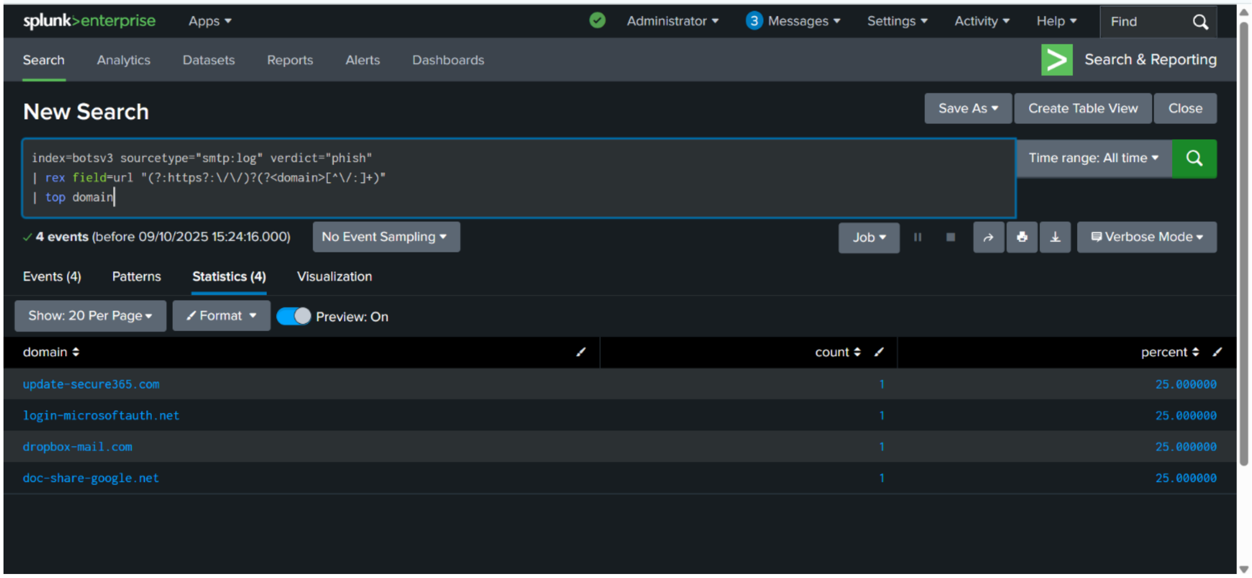
- **`index=botsv3`** → Searches the BOTS v3 dataset.
- **`sourcetype="smtp:log"`** → Restricts results to email logs.
- **`verdict="phish"`** → Filters for phishing emails only.
- **`| top recipient`** → Uses Splunk's built-in `top` command to count which recipients appear most frequently in phishing events.

Purpose

This query provides visibility into:

- Which **users or departments** are most targeted by phishing attempts.
- Whether attacks are **focused on high-value users** (e.g., executives, HR, finance).
- Where to direct **phishing awareness campaigns** or **technical protections** (like email filters or DMARC enforcement).

Common phishing domains.



```
index=botsv3 sourcetype="smtp:log" verdict="phish"
| rex field=url "(?:https?:\/\/)?(?<domain>[^\/:]+)"
| top domain
```

✓ **4 events** (before 09/10/2025 15:24:16.000)

domain	count	percent
update-secure365.com	1	25.000000
login-microsoftauth.net	1	25.000000
dropbox-mail.com	1	25.000000
doc-share-google.net	1	25.000000

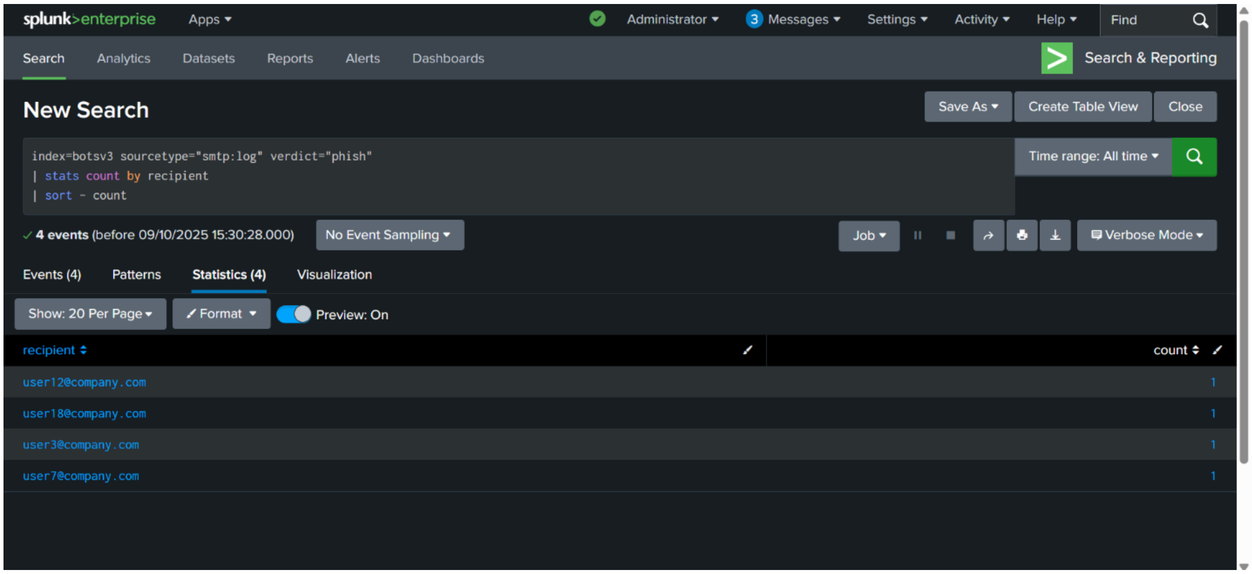
Explanation / Note

This query identifies the **most frequently used domains** in phishing URLs found in email logs from the BOTS v3 dataset.
It helps analysts uncover **malicious web infrastructure** that attackers use repeatedly in phishing campaigns.
Breakdown

- `index=botsv3` → Searches within the BOTS v3 dataset.
- `sourcetype="smtp:log"` → Focuses only on email (SMTP) logs.
- `verdict="phish"` → Filters results to only phishing emails.
- `| rex field=url "(?:https?:\/\/)?(?<domain>[^\/:]+)"` → Uses a regex (regular expression) to extract the domain name from the full URL.

4. A phishing campaign targets the HR team with Excel attachments named 'Payroll_Q4.xlsm'. Write a query to detect all recipients and their attachment hash values using sourcetype='smtp:log'.

Ans.



```
index=botsv3 sourcetype="smtp:log" verdict="phish"
| stats count by recipient
| sort - count
```

✓ **4 events** (before 09/10/2025 15:30:28.000)

recipient	count
user12@company.com	1
user18@company.com	1
user3@company.com	1
user7@company.com	1

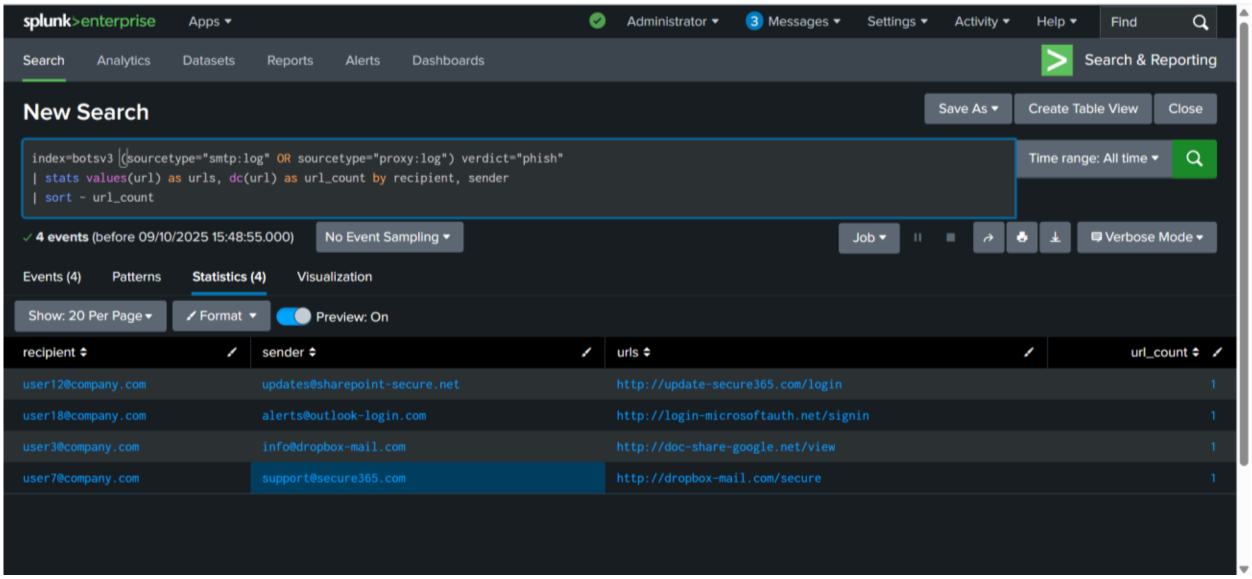
Explanation / Note

This query lists all **recipients who have received phishing emails**, sorted by the total number of phishing messages each one received.
It provides a **clear ranking of the most targeted users** in the organization.
Breakdown

- **`index=botsv3`** → Searches within the BOTS v3 dataset.
- **`sourcetype="smtp:log"`** → Restricts results to email (SMTP) logs.
- **`verdict="phish"`** → Filters for emails marked as phishing.
- **`| stats count by recipient`** →
  Uses the `stats` command to:
  - **Group** all phishing emails by each recipient (user).
  - **Count** how many phishing messages each one received.
- **`| sort - count`** → Sorts the list in descending order, so the most targeted users appear at the top.

5. The threat intel feed (sourcetype='threatintel:domains') lists 'update-secure365.com' as malicious. Create an SPL search to find if this domain appears in any email or proxy events in the past 7 days.
   Ans.

Correlate Email with Proxy Logs (Click Activity)



```
index=botsv3 (sourcetype="smtp:log" OR sourcetype="proxy:log") verdict="phish"
| stats values(url) as urls, dc(url) as url_count by recipient, sender
| sort - url_count
```

✓ **4 events** (before 09/10/2025 15:48:55.000)

recipient	sender	urls	url_count
user12@company.com	updates@sharepoint-secure.net	http://update-secure365.com/login	1
user18@company.com	alerts@outlook-login.com	http://login-microsoftauth.net/signin	1
user3@company.com	info@dropbox-mail.com	http://doc-share-google.net/view	1
user7@company.com	support@secure365.com	http://dropbox-mail.com/secure	1

This query **correlates phishing data between email and proxy logs** to determine which senders and recipients are associated with phishing URLs, and how many unique URLs are involved.
It helps identify whether users who received phishing emails may have also **interacted with phishing links** observed in web proxy logs.
Breakdown

- `index=botsv3` → Searches in the BOTS v3 dataset.
- `(sourcetype="smtp:log" OR sourcetype="proxy:log")` → Combines data from both email (SMTP) and proxy logs.
  - `smtp:log` → Captures emails that were flagged as phishing.
  - `proxy:log` → Captures web browsing or click activity that may relate to phishing URLs.
- `verdict="phish"` → Filters only phishing-related events.
- `| stats values(url) as urls, dc(url) as url_count by recipient, sender` →
  - Groups events by **recipient** (the user) and **sender** (the attacker).
  - Lists all associated **URLs** seen in these events.
  - Counts how many **unique URLs** each pair has (`dc()` = distinct count).
- `| sort - url_count` → Sorts by the number of unique phishing URLs in descending order to highlight the most active pairs.

Interpretation

The query reveals **which senders are delivering phishing URLs to which recipients**, and how many unique phishing links were involved.
This helps analysts detect **potential compromise attempts** or **click-throughs** to phishing domains.
Observations

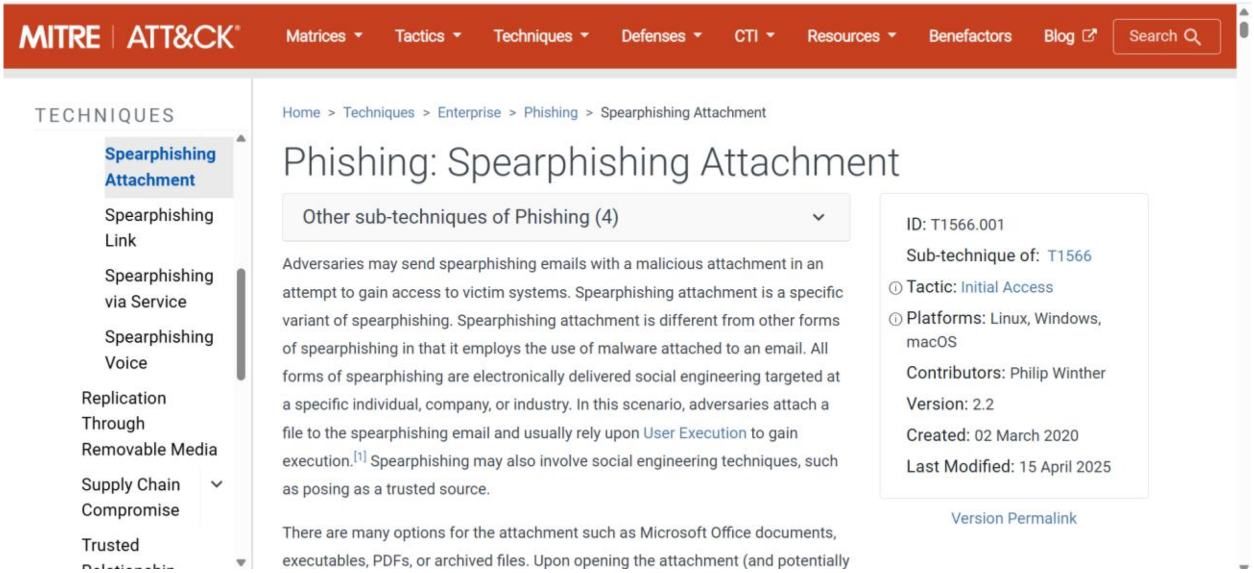
- Recipients with multiple URLs from a single sender indicate a **coordinated phishing attempt**.
- If proxy logs confirm matching URLs, this means the **user may have clicked the link**, potentially leading to endpoint compromise.
- Repeated sender-recipient pairs are **high-risk indicators** that may require deeper investigation.

🎯 Recommended Actions

- **Cross-check proxy logs** for evidence of user clicks or downloads from these URLs.
- Add these phishing domains to **blocklists or email filters**.
- Notify the affected users and initiate **incident response** procedures if necessary.
- Enrich sender and domain indicators with **threat intelligence** feeds to determine if they are part of known phishing campaigns.

11. MITRE ATT&CK mapping (quick)

- Initial Access: Phishing (T1566).



Home > Techniques > Enterprise > Phishing > Spearphishing Attachment

Phishing: Spearphishing Attachment

Other sub-techniques of Phishing (4)

Adversaries may send spearphishing emails with a malicious attachment in an attempt to gain access to victim systems. Spearphishing attachment is a specific variant of spearphishing. Spearphishing attachment is different from other forms of spearphishing in that it employs the use of malware attached to an email. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email and usually rely upon User Execution to gain execution.[1] Spearphishing may also involve social engineering techniques, such as posing as a trusted source.

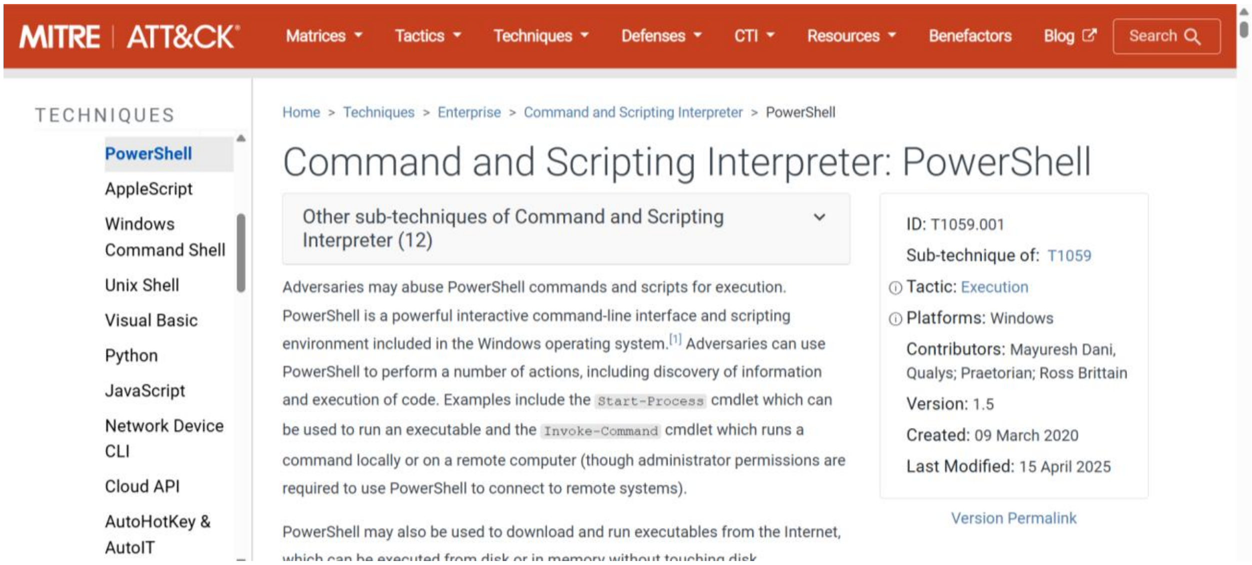
There are many options for the attachment such as Microsoft Office documents, executables, PDFs, or archived files. Upon opening the attachment (and potentially

ID: T1566.001
Sub-technique of: T1566
ⓘ Tactic: Initial Access
ⓘ Platforms: Linux, Windows, macOS
Contributors: Philip Winther
Version: 2.2
Created: 02 March 2020
Last Modified: 15 April 2025

Version Permalink

Attackers often gain entry through phishing emails containing malicious attachments or links. In this project, suspicious emails (with subjects like *invoice*, *payment*, *urgent*) were analyzed from the dataset to identify potential phishing attempts.

- Execution: PowerShell (T1059.001).



Home > Techniques > Enterprise > Command and Scripting Interpreter > PowerShell

Command and Scripting Interpreter: PowerShell

Other sub-techniques of Command and Scripting Interpreter (12)

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.[1] Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).

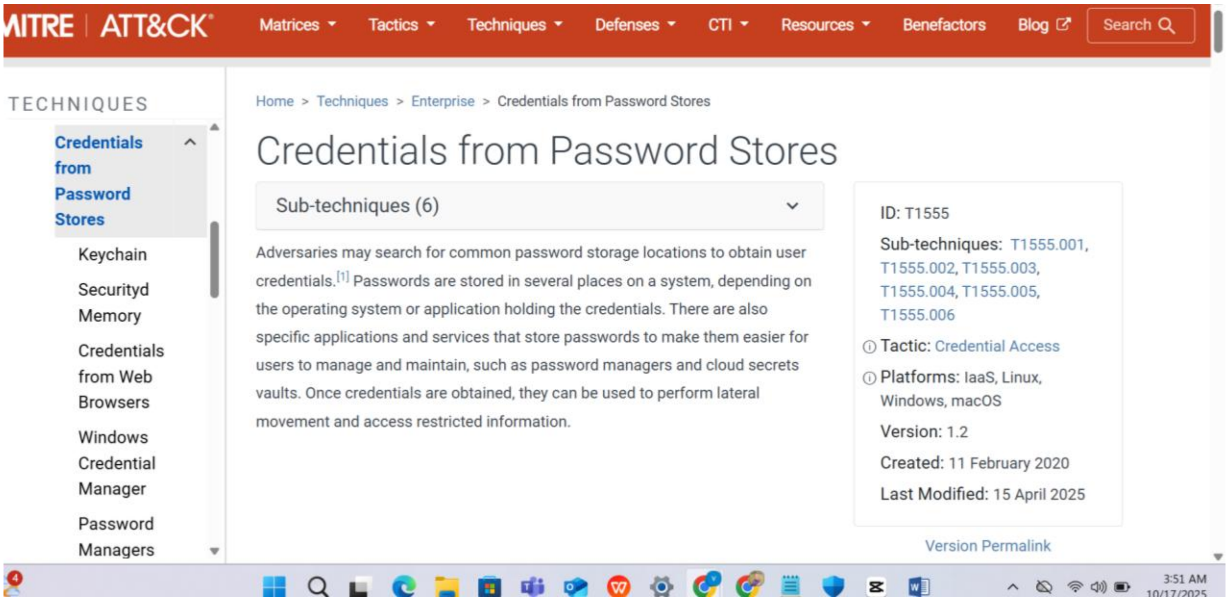
PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk

ID: T1059.001
Sub-technique of: T1059
ⓘ Tactic: Execution
ⓘ Platforms: Windows
Contributors: Mayuresh Dani, Qualys; Praetorian; Ross Brittain
Version: 1.5
Created: 09 March 2020
Last Modified: 15 April 2025

Version Permalink

Once access is gained, attackers frequently execute malicious PowerShell commands to download or execute payloads.
The PowerShell logs from the dataset were inspected for suspicious command-line activities and encoded scripts.

- Credential Access: Credentials from Web Forms / Input Capture



Home > Techniques > Enterprise > Credentials from Password Stores

Credentials from Password Stores

Sub-techniques (6)

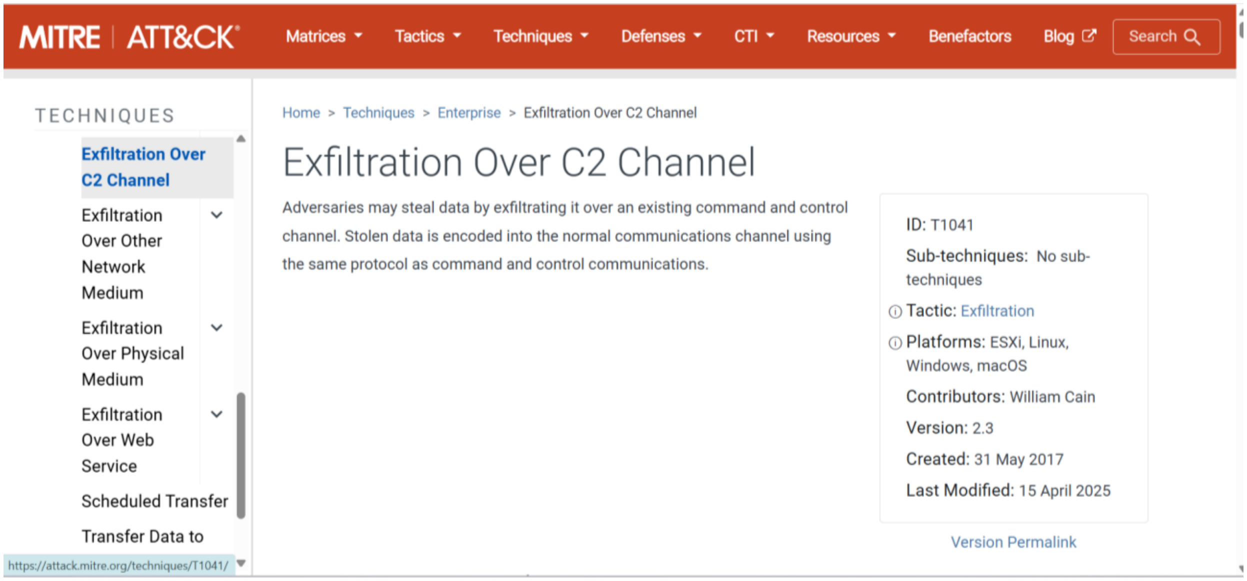
Adversaries may search for common password storage locations to obtain user credentials.[1] Passwords are stored in several places on a system, depending on the operating system or application holding the credentials. There are also specific applications and services that store passwords to make them easier for users to manage and maintain, such as password managers and cloud secrets vaults. Once credentials are obtained, they can be used to perform lateral movement and access restricted information.

ID: T1555
Sub-techniques: T1555.001, T1555.002, T1555.003, T1555.004, T1555.005, T1555.006
ⓘ Tactic: Credential Access
ⓘ Platforms: IaaS, Linux, Windows, macOS
Version: 1.2
Created: 11 February 2020
Last Modified: 15 April 2025

Version Permalink

Adversaries may attempt to steal credentials through fake login pages or by intercepting form submissions.
HTTP POST requests containing parameters like *username* or *password* were analyzed to identify potential credential harvesting attempts.

- Command & Control: Exfiltration over HTTP (T1041 / T1071.001).


MITRE | ATT&CK®
Matrices
Tactics
Techniques
Defenses
CTI
Resources
Benefactors
Blog
Search
TECHNIQUES
Exfiltration Over C2 Channel
Exfiltration Over Other Network Medium
Exfiltration Over Physical Medium
Exfiltration Over Web Service
Scheduled Transfer
Transfer Data to
https://attack.mitre.org/techniques/T1041/

Home > Techniques > Enterprise > Exfiltration Over C2 Channel

Exfiltration Over C2 Channel

Adversaries may steal data by exfiltrating it over an existing command and control channel. Stolen data is encoded into the normal communications channel using the same protocol as command and control communications.

ID: T1041

Sub-techniques: No sub-techniques

Tactic: Exfiltration

Platforms: ESXi, Linux, Windows, macOS

Contributors: William Cain

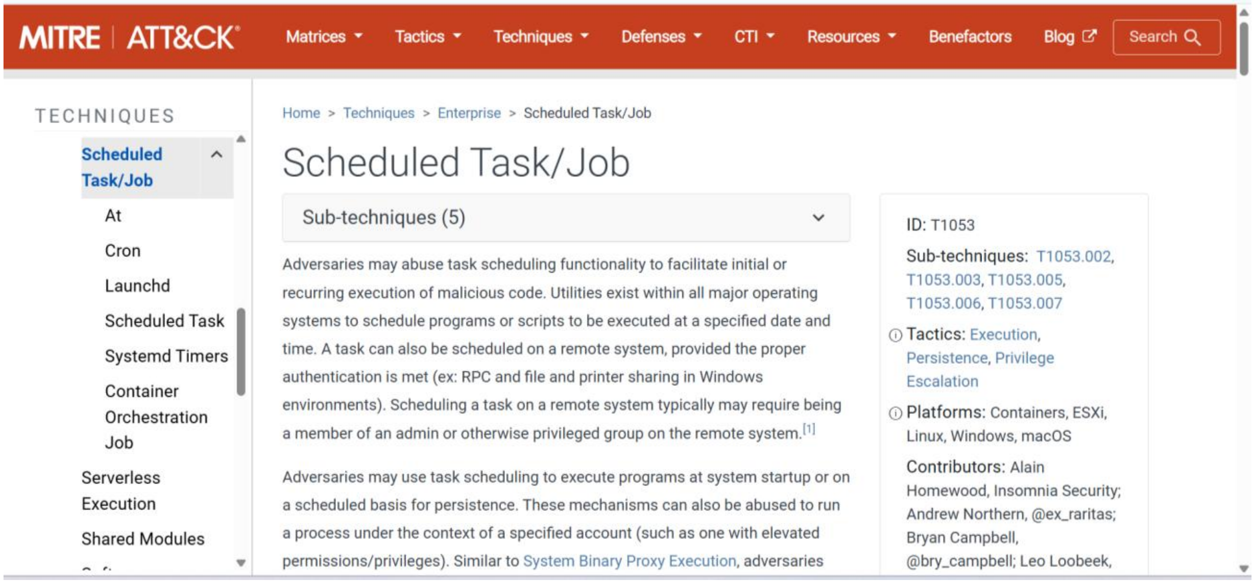
Version: 2.3

Created: 31 May 2017

Last Modified: 15 April 2025

Version Permalink

- Persistence: Scheduled Task / Registry (T1053 / T1112) — check for after-discover



Home > Techniques > Enterprise > Scheduled Task/Job

Scheduled Task/Job

Sub-techniques (5)

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.[1]

Adversaries may use task scheduling to execute programs at system startup or on a scheduled basis for persistence. These mechanisms can also be abused to run a process under the context of a specified account (such as one with elevated permissions/privileges). Similar to System Binary Proxy Execution, adversaries

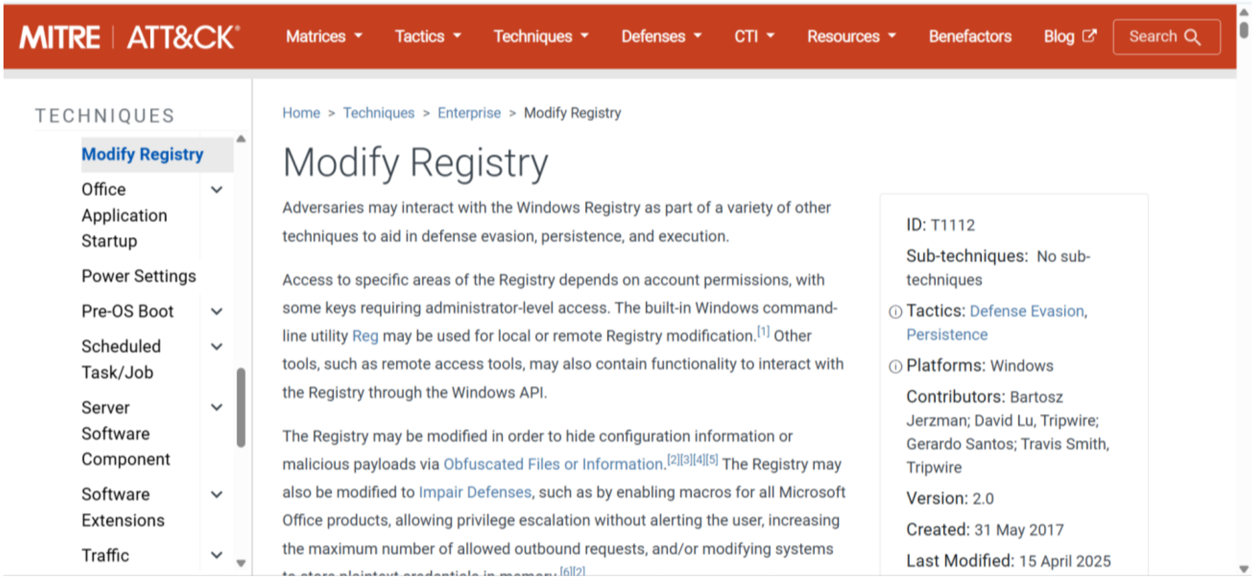
ID: T1053

Sub-techniques: T1053.002, T1053.003, T1053.005, T1053.006, T1053.007

Tactics: Execution, Persistence, Privilege Escalation

Platforms: Containers, ESXi, Linux, Windows, macOS

Contributors: Alain Homewood, Insomnia Security; Andrew Northern, @ex_raritas; Bryan Campbell, @bry_campbell; Leo Loobeek,



Home > Techniques > Enterprise > Modify Registry

Modify Registry

Adversaries may interact with the Windows Registry as part of a variety of other techniques to aid in defense evasion, persistence, and execution.

Access to specific areas of the Registry depends on account permissions, with some keys requiring administrator-level access. The built-in Windows command-line utility Reg may be used for local or remote Registry modification.[1] Other tools, such as remote access tools, may also contain functionality to interact with the Registry through the Windows API.

The Registry may be modified in order to hide configuration information or malicious payloads via Obfuscated Files or Information.[2][3][4][5] The Registry may also be modified to Impair Defenses, such as by enabling macros for all Microsoft Office products, allowing privilege escalation without alerting the user, increasing the maximum number of allowed outbound requests, and/or modifying systems to store plaintext credentials in memory.[6][2]

ID: T1112

Sub-techniques: No sub-techniques

ⓘ Tactics: Defense Evasion, Persistence

ⓘ Platforms: Windows

Contributors: Bartosz Jerzman; David Lu, Tripwire; Gerardo Santos; Travis Smith, Tripwire

Version: 2.0

Created: 31 May 2017

Last Modified: 15 April 2025

Note:

To maintain access, attackers often create scheduled tasks or modify registry run keys. Windows event logs were investigated for registry persistence and suspicious task creation events. 🛡 □ Quick Hardening & Detection Tips

1. Enforce Multi-Factor Authentication (MFA)
Ensure MFA is enabled for all privileged and external-facing accounts to limit the impact of stolen credentials during phishing attacks.

2. Restrict PowerShell Execution
Use **Constrained Language Mode**, **AppLocker**, or **Windows Defender Application Control (WDAC)** to block unauthorized PowerShell scripts.
Only allow signed scripts from approved administrators.

3. Harden Microsoft Office Settings
Disable Office macros by default and flag attachments containing `.vbs`, `.js`, or `.exe` scripts as high risk.
Encourage users to open attachments in sandboxed or protected environments.

4. Integrate Threat Intelligence Feeds
Connect external intelligence sources like **PhishTank**, **VirusTotal**, or **AbuseIPDB** into Splunk lookups to automatically enrich domain, IP, or hash detections.

5. Baseline Normal Network Behavior
In Splunk, build a baseline of normal HTTP POST destinations for internal applications.
Alert on new or rare external domains receiving POST requests — a common sign of credential exfiltration.