Attack Conditions

The provenance queries are constructed from two queries (prerequisite query and the main query). Here, we are consolidating the conditions for these two building blocks queries for different attack behaviors.

Initial Compromise.
Domain Hijaking (T1584-001).
The prerequisite query:

$$N_1 \in \{BrowserProcesses\} \;\wedge\; R_I = fork \wedge \\ N_2 \in \{RemoteAccessProcesses\}$$

The main query:

$$N_3 \in \{N_2\} \;\wedge R_M = connect \wedge \\ N_4.ip \notin \{TrustedIPAddresses\}$$

where *RemoteAccessProcesses* is the list of processes used for remote access (e.g., SSHD).
Exploit Public-Facing Applications (T1190).
The prerequisite query:

$$N_1 \in \{PublicFacingProcesses\} \;\wedge R_I = accept \wedge \\ N_2.ip \notin \{TrustedIPAddresses\}$$

The main query:

$$N_3 \in \{PublicFacingProcesses\} \;\wedge\; R_M = connect \wedge \\ N_4.ip \notin \{TrustedIPAddresses\} \;\wedge N_4 \notin \{N_2\}$$

where N_2 and N_4 represent sockets (IP Address and Port).

Non Standard Port (T1571).
The prerequisite query: None
The main query:

$$N_4.ip \notin \{TrustedIPAddresses\} \wedge \\ Pair(Process\ N_3\ and\ Port\ N_4.port) \notin \\ \{ServicePortList\} \;\wedge\; R_M = connect$$

Establish Foothold.
The prerequisite query:

$$N_1 \in \{CompromisedProcesses\} \wedge \\ R_I \in \{fork, execute\} \wedge \\ length \leq SelectedLength$$

CompromisedProcesses is a set of all processes tagged as compromised from the Initial Compromise stage.
The main query:

$$N_3 \in \{N_2\} \;\wedge\; R_M \in \{fork, execute\} \wedge \\ N_4 \in \{CommandLineUtilities\}$$

Escalate Privileges.
Super User Privilege.
The prerequisite query:

$$N_1 \in \{CompromisedProcesses\} \wedge \\ R_I \in \{fork, execute\} \wedge \\ length \leq SelectedLength$$

The main query:

$$N_3 \in \{N_2\} \;\wedge\; R_M \in \{ChangePrincipal\} \wedge \\ N_4.uid \in \{SuperUsers\}$$

ChangePrincipal is the set of syscalls that change the owner user. Examples of those syscalls are `chown`, `fchown`, and `lchown` syscalls. $N_4.uid$ is the real user ID of the affected process (N_4).

Super User Utilities.
The prerequisite query:

$$N_1 \in \{CompromisedProcesses\} \wedge \\ R_I \in \{fork, execute\} \wedge \\ length \leq SelectedLength$$

The main query:

$$N_3 \in \{N_2\} \;\wedge\; R_M \in \{fork, execute\} \wedge \\ N_4 \in \{SuperUserUtilities\}$$

Scheduled Tasks.

The prerequisite query:

$$N_1 \in \{cron.d\} \wedge \\ R_I \in \{fork, execute\} \wedge \\ length \leq SelectedLength$$

The main query:

$$N_3 \in \{N_2\} \;\wedge\; R_M \in \{chown\} \wedge \\ N_4.uid \in \{SuperUsers\}$$

Credential Dump.
The prerequisite query:

$$R_I \in \{fork, execute\} \;\wedge\; N_1 \in \\ (\{CompromisedProcesses\} \;\vee\; \{SuperUserPrivilege\}) \\ \wedge\; length \leq SelectedLength$$

SuperUserPrivilege is a set of all processes tagged as compromised from *Super User Privilege* technique in Escalate Privileges stage.
The main query:

$$N_3 \in \{N_2\} \;\wedge\; N_4.path\ contains\ “procdump'' \wedge \\ (N_3.euid \in \{SuperUsers\} \;\vee N_3.uid \in \{SuperUsers\}) \wedge \\ R_M = execute$$

$N_3.uid$ and $N_3.euid$ are the real and effective user IDs for the process N_3 respectively.

Valid Domain Accounts.
The prerequisite query:

$$N_1.uid \notin \{DomainUsers\} \wedge \\ N_2 \in \{ScriptingProcesses\} \wedge \\ R_I \in \{fork, execute\} \wedge \\ length \leq SelectedLength$$

The main query:

$$\begin{gathered} N_3 \in \{N_2\} \;\wedge\; R_M = connect \wedge \\ N_3.uid \in \{DomainUsers\} \wedge \\ N_4 \in \{InternalIPAddresses\} \wedge \\ N_4.ip \in \{DomainIPAddresses\} \end{gathered}$$

DomainUsers is the list of users who are authorized to access the domain controller. *ScriptingProcesses* examples include Python and Powershell. *DomainIPAddresses* are IP addresses of the domain controllers. *uid* is the user id for the corresponding process.

Internal Reconnaissance.

Sensitive Access.

The prerequisite query:

$$\begin{aligned} & N_1 \in \{CompromisedProcesses\} \wedge \\ & R_I \in \{fork, execute\} \wedge \\ & length \leq SelectedLength \end{aligned}$$

The prerequisite query:

$$\begin{gathered} N_3 \in \{N_2\} \;\wedge\; R_M \in \{open, read\} \;\wedge\; N_4 \in \\ (\{SensitivePaths\} \vee \{SystemCriticalPaths\}) \end{gathered}$$

Recon Command.

The prerequisite query:

$$\begin{aligned} & N_1 \in \{CompromisedProcesses\} \wedge \\ & R_I \in \{fork, execute\} \wedge \\ & length \leq SelectedLength \end{aligned}$$

The main query:

$$\begin{aligned} & N_3 \in \{N_2\} \;\wedge\; R_M = execute \wedge \\ & N_4 \in \{SensitiveCommands\} \end{aligned}$$

Port Scan.

The prerequisite query:

$$\begin{aligned} & N_1 \in \{CompromisedProcesses\} \wedge \\ & R_I \in \{fork, execute\} \wedge \\ & length \leq SelectedLength \end{aligned}$$

The main query:

$$\begin{gathered} N_3 \in \{N_2\} \;\wedge\; R_M = send \wedge \\ N_4 \in \{InternalIPAddresses\} \wedge \\ N_4.port \in \{WellKnownPorts\} \wedge \\ count(N_4.port) \geq PortCountThres \end{gathered}$$

Based on the enterprise settings, the analyst selects the number of ports ($PortCountThres$) at which the provenance query should generate an alert.

Lateral Movement.

The prerequisite query:

$$\begin{aligned} & (N_1 \in \{CompromisedProcesses\} \vee \\ & N_1 \in \{InternalReconProcesses\}) \wedge \\ & R_I \in \{fork, execute\} \wedge \\ & length \leq SelectedLength \end{aligned}$$

The main query:

$$\begin{aligned} & N_3 \in \{N_2\} \;\wedge\; R_M = connect \wedge \\ & N_4 \in \{InternalIPAddresses\} \end{aligned}$$

InternalReconProcesses is a set of all processes engaged in Internal Reconnaissance activities.

Complete Mission.

Exfiltration Over C2 Channel.

The prerequisite query:

$$\begin{aligned} & (N_1 \in \{CompromisedProcesses\} \vee \\ & N_1 \in \{InternalReconProcesses\}) \wedge \\ & R_I \in \{fork, execute\} \wedge \\ & length \leq SelectedLength \end{aligned}$$

The main query:

$$\begin{aligned} & N_3 \in \{N_2\} \wedge N_4 \notin \{TrustedIPAddresses\} \wedge \\ & R_M = send \end{aligned}$$

`send` is the family of syscalls that includes syscalls used to send data over the network including sendmsg, sendto, sendfile, etc.

Exfiltration by Bypassing Defense Controls.

The prerequisite query:

$$\begin{aligned} & N_1 \in \{InternalReconProcesses\} \wedge \\ & \quad \in \{EscalatePrivilegeProcesses\} \wedge \\ & R_I \in \{fork, execute\} \\ & \wedge \;\; length \leq SelectedLength \end{aligned}$$

The main query:

$$\begin{aligned} & N_3 \in \{N_2\} \;\wedge N_3.uid \in \{SuperUsers\} \wedge \\ & N_4 \notin \{TrustedIPAddresses\} \;\wedge R_M = send \end{aligned}$$

EscalatePrivilegeProcesses is a set of all processes with super user privileges detected in Escalate Privileges stage.

Destroy System.

The prerequisite query:

$$\begin{aligned} & (N_1 \in \{CompromisedProcesses\} \vee \\ & N_1 \in \{InternalReconProcesses\}) \wedge \\ & R_I \in \{fork, execute\} \wedge \\ & length \leq SelectedLength \end{aligned}$$

The main query:

$$\begin{aligned} N_3 \;& \in \{N_2\} \wedge \\ (N_4 \;& \in \{SensitivePaths\} \vee \\ & \in \{SystemFiles\}) \wedge \\ R_M \;& \in \{write, unlink\} \end{aligned}$$

SystemFiles can be generic per the operating system and can be based on the CTI report. *SensitivePaths* is customised per every enterprise. Here, enterprises define paths to sensitive files and directories. This includes user drives, internal, confidential, and secret shares. Any suspicious operation on those files will be flagged and an alert will be generated.

Cleanup Tracks.

File Deletion.

The prerequisite query:

$$\begin{aligned} & N_1 \in (\{CompromisedProcesses\} \vee \\ & \quad \in \{EscalatePrivilegeProcesses\}) \wedge \\ & R_I \in \{fork, execute\} \wedge \\ & \quad length \leq SelectedLength \end{aligned}$$

The main query:

$$N_3 \ \in \ \{N_2\} \ \wedge \ N_4 \ \notin \ \{LogFilesPaths\} \wedge$$
$$R_M \ = \ unlink$$

Remove Log files.

The prerequisite query:

$$N_1 \in (\{CompromisedProcesses\} \vee$$
$$\in \{EscalatePrivilegeProcesses\}) \wedge$$
$$R_I \in \{fork, execute\} \wedge$$
$$length \leq SelectedLength$$

The main query:

$$N_3 \ \in \ \{N_2\} \ \wedge \ N_4 \ \in \ \{LogFilesPaths\} \wedge$$
$$R_M \ = \ unlink$$

Clear Log commands.

The prerequisite query:

$$N_1 \in (\{CompromisedProcesses\} \vee$$
$$\in \{EscalatePrivilegeProcesses\}) \wedge$$
$$R_I \in \{fork, execute\} \wedge$$
$$length \leq SelectedLength$$

The main query:

$$N_3 \ \in \ \{N_2\} \ \wedge \ R_M = write \wedge$$
$$N_4 \ \in \ \{LogFilesPaths\}$$