CE 815 – Secure Software Systems

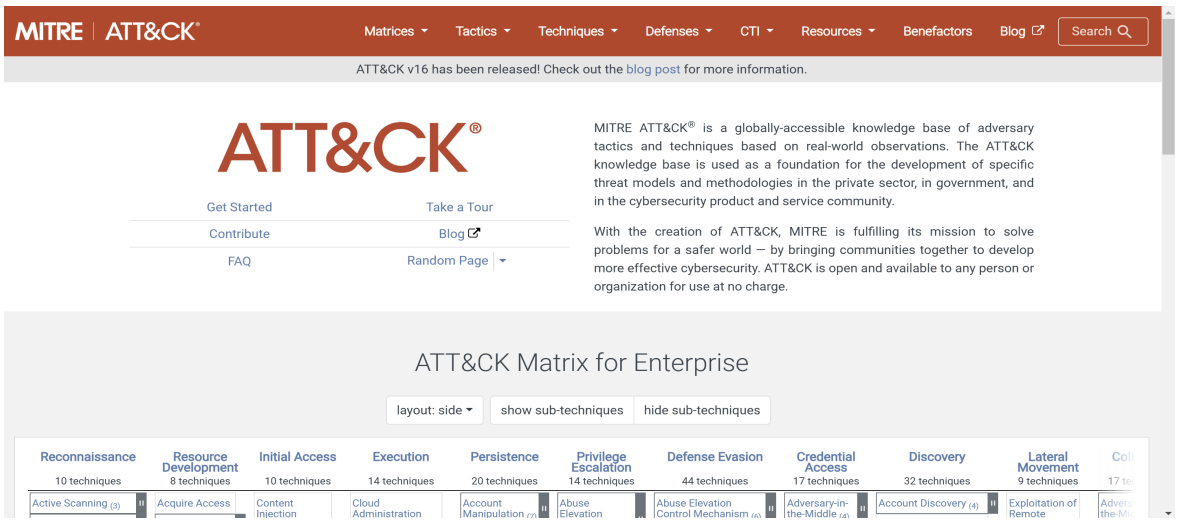
Causal Analysis (Holmes)

Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology

Acknowledgments: Some of the slides are fully or partially obtained from other sources. A reference is noted on the bottom of each slide, when the content is fully obtained from another source. Otherwise a full list of references is provided on the last slide. Thanks to Zahra Fazli for the help on the slides.

MITRE

- globally-accessible knowledge base of adversary tactics and techniques based on real-world observations



Enterprise Tactics

ID	Name	Description
TA0043	Reconnaissance	The adversary is trying to gather information they can use to plan future operations.
TA0042	Resource Development	The adversary is trying to establish resources they can use to support operations.
TA0001	Initial Access	The adversary is trying to get into your network.
TA0002	Execution	The adversary is trying to run malicious code.
TA0003	Persistence	The adversary is trying to maintain their foothold.
TA0004	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0005	Defense Evasion	The adversary is trying to avoid being detected.
TA0006	Credential Access	The adversary is trying to steal account names and passwords.
TA0007	Discovery	The adversary is trying to figure out your environment.
TA0008	Lateral Movement	The adversary is trying to move through your environment.
TA0009	Collection	The adversary is trying to gather data of interest to their goal.
TA0011	Command and Control	The adversary is trying to communicate with compromised systems to control them.
TA0010	Exfiltration	The adversary is trying to steal data.
TA0040	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

Enterprise Techniques: Initial Access

ID	Name	Description
T1659	Content Injection	Adversaries may gain access and continuously communicate with victims by injecting malicious content into systems through online network traffic. Rather than luring victims to malicious payloads hosted on a compromised website (i.e., Drive-by Target followed by Drive-by Compromise), adversaries may initially access victims through compromised data-transfer channels where they can manipulate traffic and/or inject their own content. These compromised online network channels may also be used to deliver additional payloads (i.e., Ingress Tool Transfer) and other data to already compromised systems.
T1189	Drive-by Compromise	Adversaries may gain access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is typically targeted for exploitation, but adversaries may also use compromised websites for non-exploitation behavior such as acquiring Application Access Token.
T1190	Exploit Public-Facing Application	Adversaries may attempt to exploit a weakness in an Internet-facing host or system to initially access a network. The weakness in the system can be a software bug, a temporary glitch, or a misconfiguration.
T1133	External Remote Services	Adversaries may leverage external-facing remote services to initially access and/or persist within a network. Remote services such as VPNs, Citrix, and other access mechanisms allow users to connect to internal enterprise network resources from external locations. There are often remote service gateways that manage connections and credential authentication for these services. Services such as Windows Remote Management and VNC can also be used externally.
T1200	Hardware Additions	Adversaries may introduce computer accessories, networking hardware, or other computing devices into a system or network that can be used as a vector to gain access. Rather than just connecting and distributing payloads via removable storage (i.e. Replication Through Removable Media), more robust hardware additions can be used to introduce new functionalities and/or features into a system that can then be abused.
T1566	Phishing	Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns.
.001	Spearphishing Attachment	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. Spearphishing may also involve social engineering techniques, such as posing as a trusted source.
.002	Spearphishing Link	Adversaries may send spearphishing emails with a malicious link in an attempt to gain access to victim systems. Spearphishing with a link is a specific variant of spearphishing. It is different from other forms of spearphishing in that it employs the use of links to download malware contained in email, instead of attaching malicious files to the email itself, to avoid defenses that may inspect email attachments. Spearphishing may also involve social engineering techniques, such as posing as a trusted source.
.003	Spearphishing via Service	Adversaries may send spearphishing messages via third-party services in an attempt to gain access to victim systems. Spearphishing via service is a specific variant of spearphishing. It is different from other forms of spearphishing in that it employs the use of third party services rather than directly via enterprise email channels.

Techniques :Content Injection

Content Injection

Adversaries may gain access and continuously communicate with victims by injecting malicious content into systems through online network traffic. Rather than luring victims to malicious payloads hosted on a compromised website (i.e., Drive-by Target followed by Drive-by Compromise), adversaries may initially access victims through compromised data-transfer channels where they can manipulate traffic and/or inject their own content. These compromised online network channels may also be used to deliver additional payloads (i.e., Ingress Tool Transfer) and other data to already compromised systems.[1]

Mitigations

ID	Mitigation	Description
M1041	Encrypt Sensitive Information	Where possible, ensure that online traffic is appropriately encrypted through services such as trusted VPNs.
M1021	Restrict Web-Based Content	Consider blocking download/transfer and execution of potentially uncommon file types known to be used in adversary campaigns.

Detection

ID	Data Source	Data Component	Detects
DS0022	File	File Creation	Monitor for unexpected and abnormal file creations that may indicate malicious content injected through online network communications.
DS0029	Network Traffic	Network Traffic Content	Monitor for other unusual network traffic that may indicate additional malicious content transferred to the system. Use network intrusion detection systems, sometimes with SSL/TLS inspection, to look for known malicious payloads, content obfuscation, and exploit code.
DS0009	Process	Process Creation	Look for behaviors on the endpoint system that might indicate successful compromise, such as abnormal behaviors of browser processes. This could include suspicious files written to disk, evidence of Process Injection for attempts to hide execution, or evidence of Discovery.

HOLMES: Real-time APT Detection through Correlation of Suspicious Information Flows, Sadegh M. Milajerdi, Rigel Gjomemo, Birhanu Eshetey, R. Sekarz, V.N. Venkatakrishnan, IEEE Symposium on Security and Privacy, 2019.

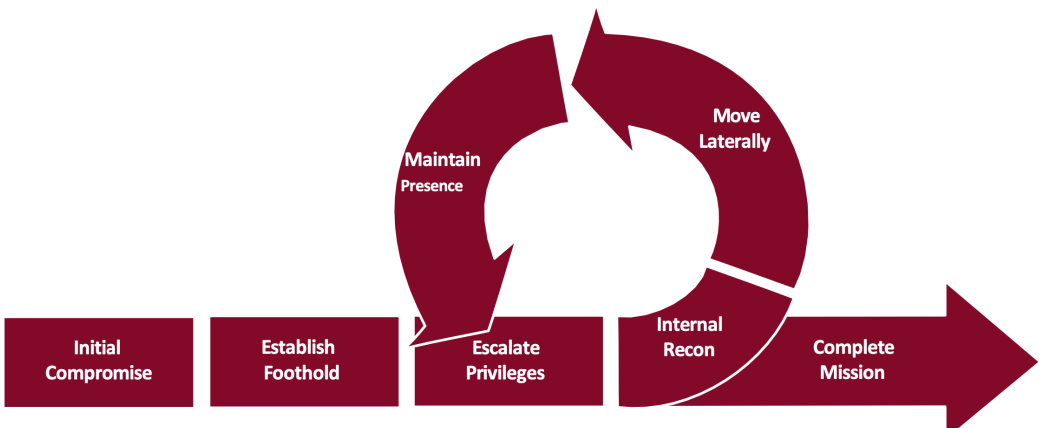
Advanced Persistent Threat (APT) and its challenges

- Targeted cyber attacks on organizations getting more sophisticated and stealthy.
- Goal: to steal data, disrupt operations or destroy infrastructure.
- APTs combine many different attack vectors
- Each appearing in some log sources
- Firewall, IDS/IPS, Netflow, DNS logs, Identity and access management tools
- Might occur over a long duration
- Correlating heterogeneous alarms using heuristics like timestamp is not so effective
- Lacking the full picture (root cause, affected entities, etc.).
- Significant manual effort and expertise are needed to piece together numerous alarms emanated by multiple security tools.

Fall 1403

Intuition

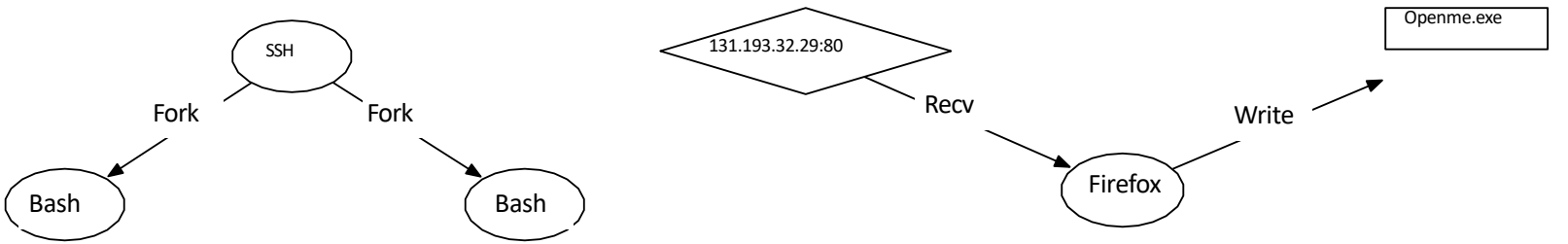
- APT behaviors often conform to the kill-chain [MANDIANT-APT1]



- Our analysis of over 300 APT whitepapers confirms that most APTs follow this kill-chain
- In particular, high-level steps of APTs need to be causally connected
- Use connectedness of high-level steps as a basis for campaign detection

Approach

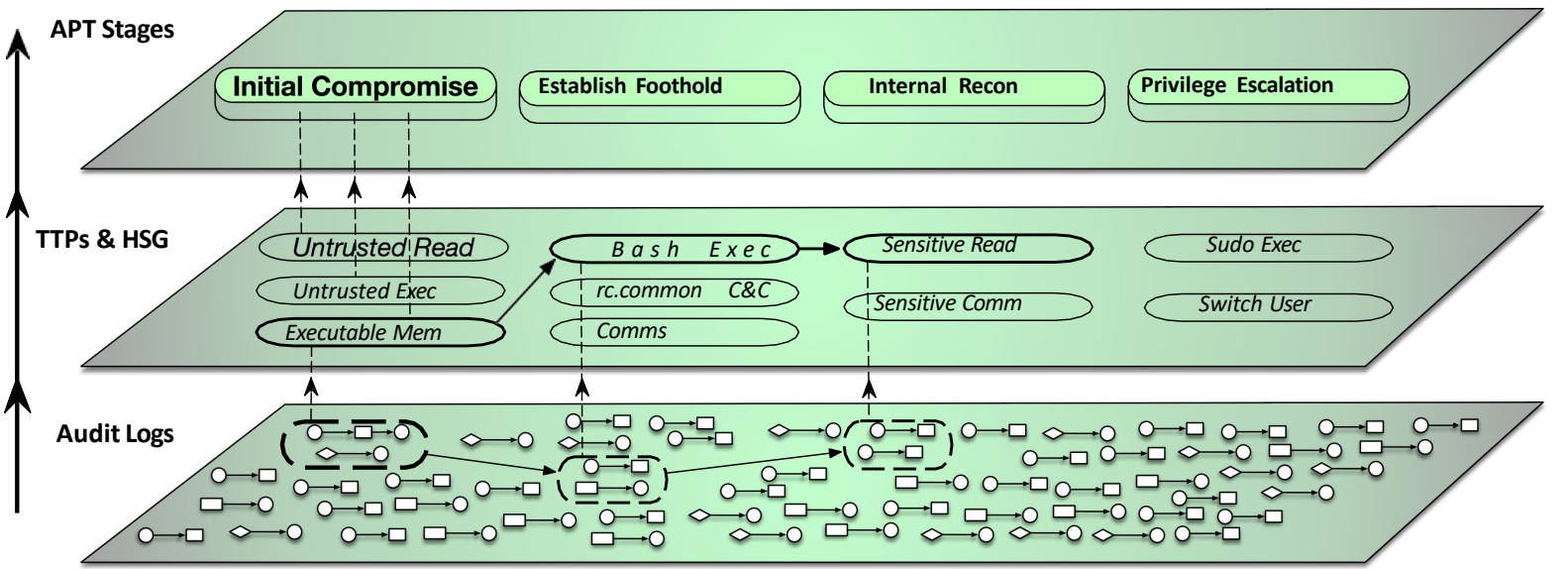
- Use Provenance Graph to enable alert correlation for attack campaign detection
- vertices: system entities (socket, process, file, memory, etc.), and agents (user, groups,…)
- edges: system calls (causal dependencies or information flow)



- Leverage the full historical context of a system
- Reason about interrelationships between different events and objects
- Key challenge: How to bridge semantic gap between low-level records and high-level activities in kill-chain?

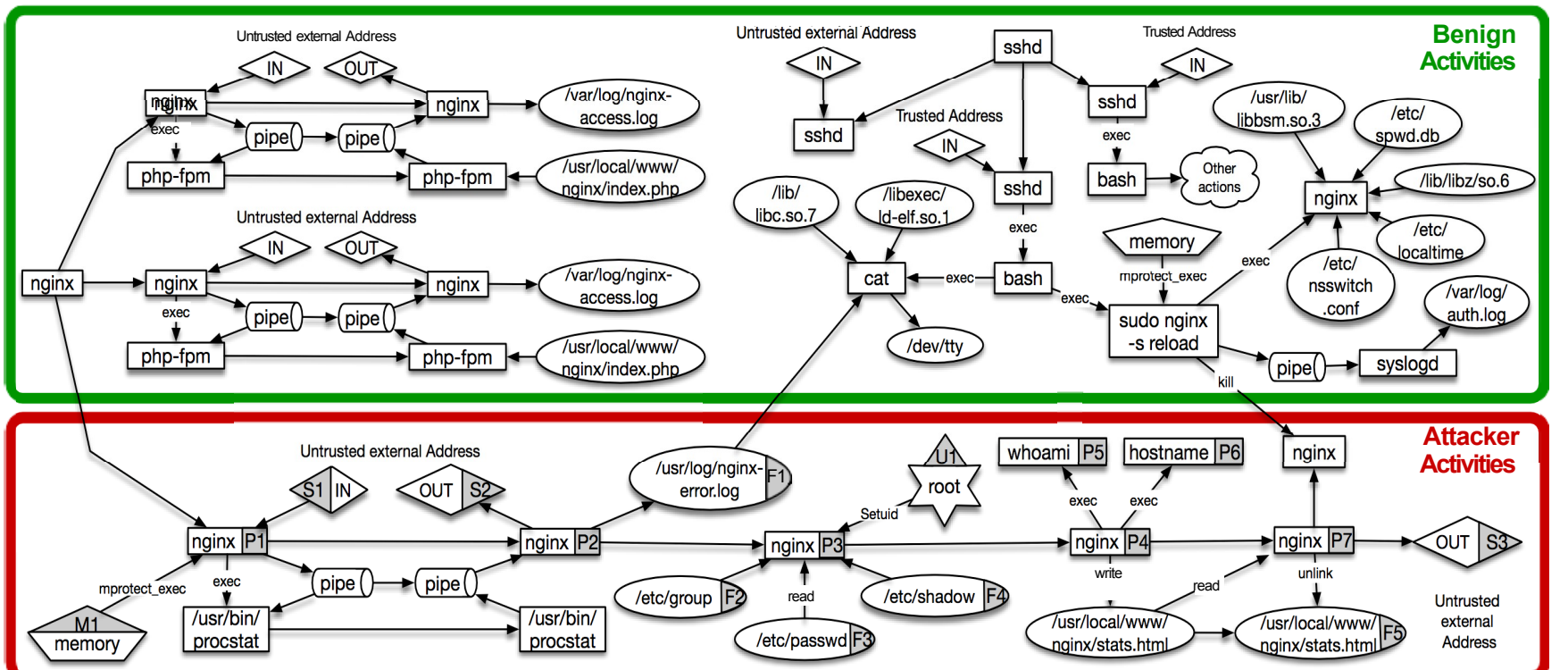
Bridging the Semantic Gap

- Use Tactics, Techniques, and Procedures (TTPs) from MITRE's ATT&CK framework as an intermediate layer to bridge low-level audit records to high-level steps

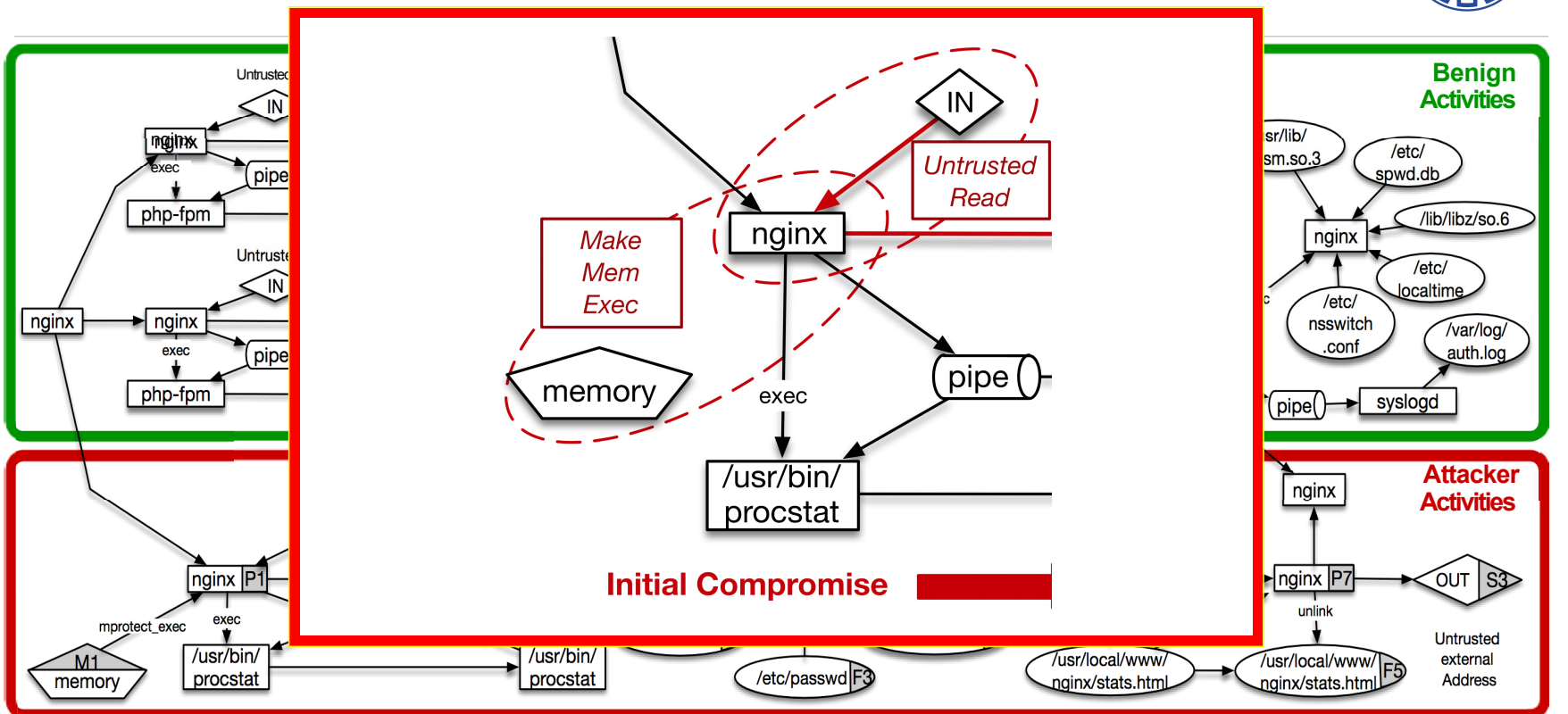


Bridging the Semantic Gap

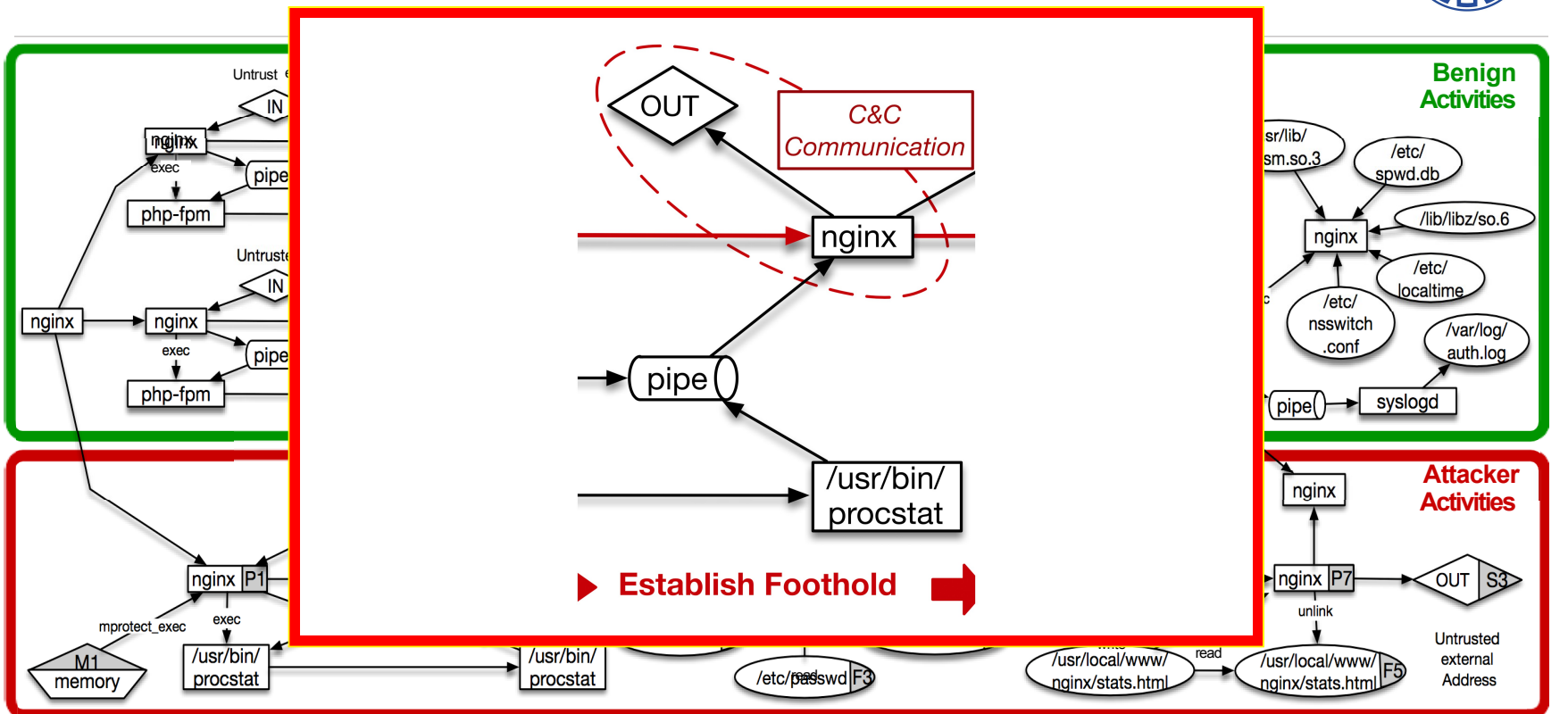
Illustrative Example



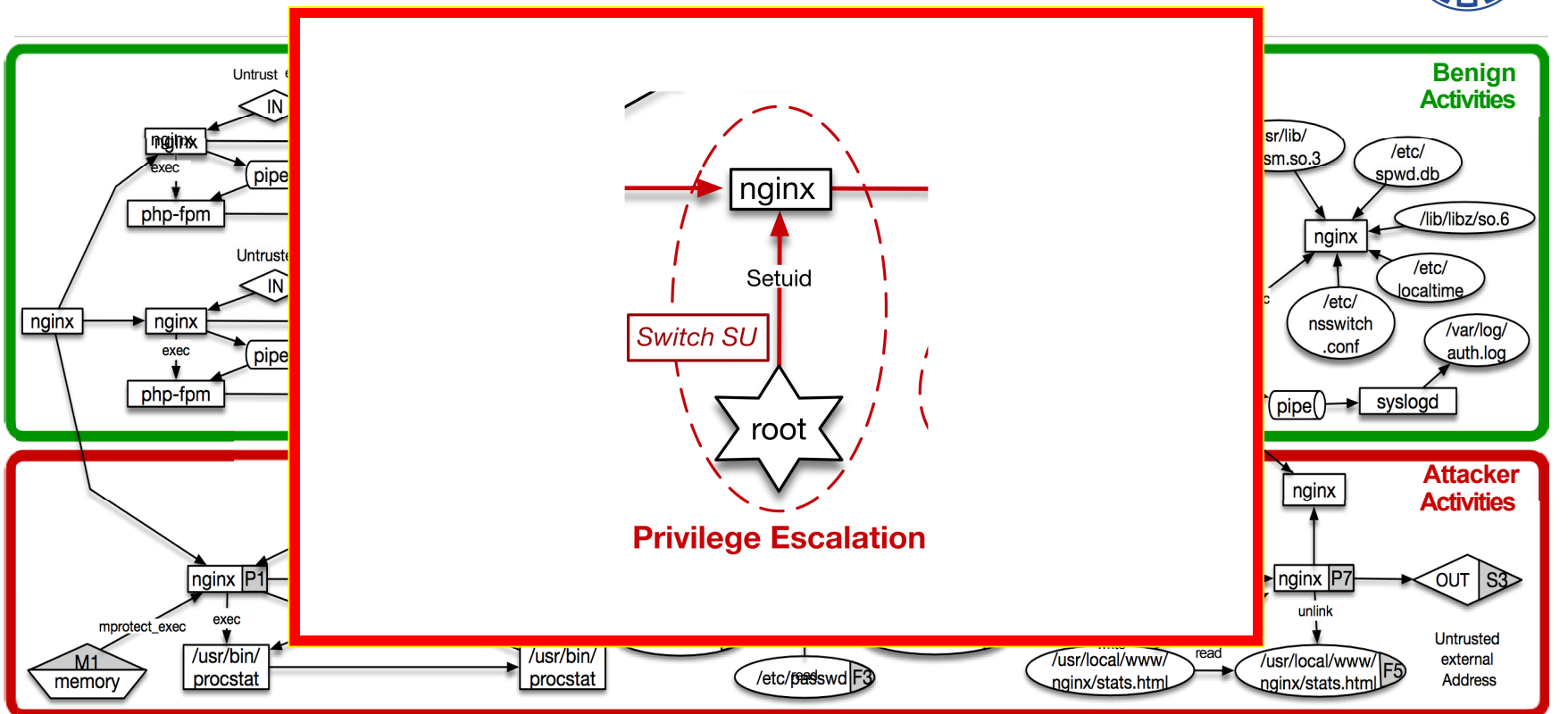
Illustrative Example



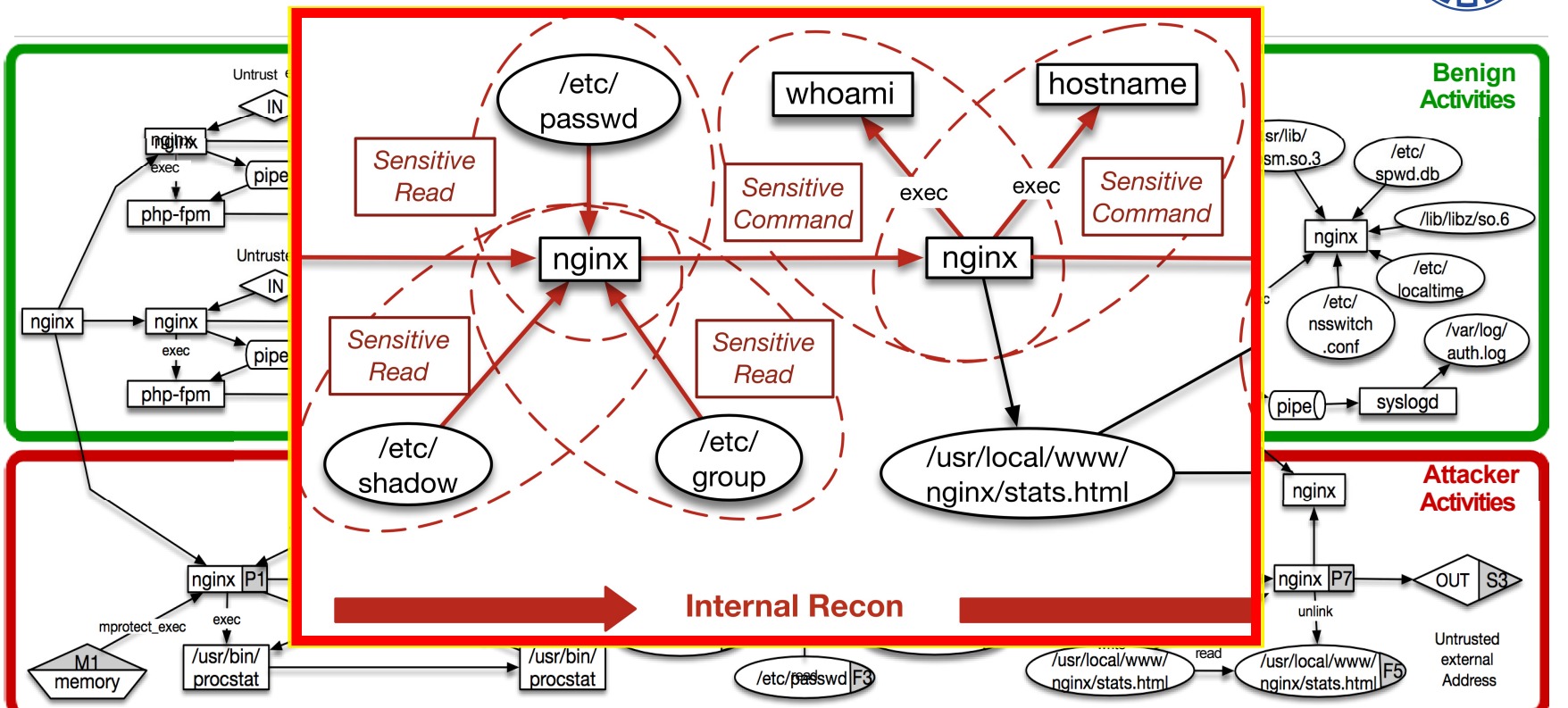
Illustrative Example



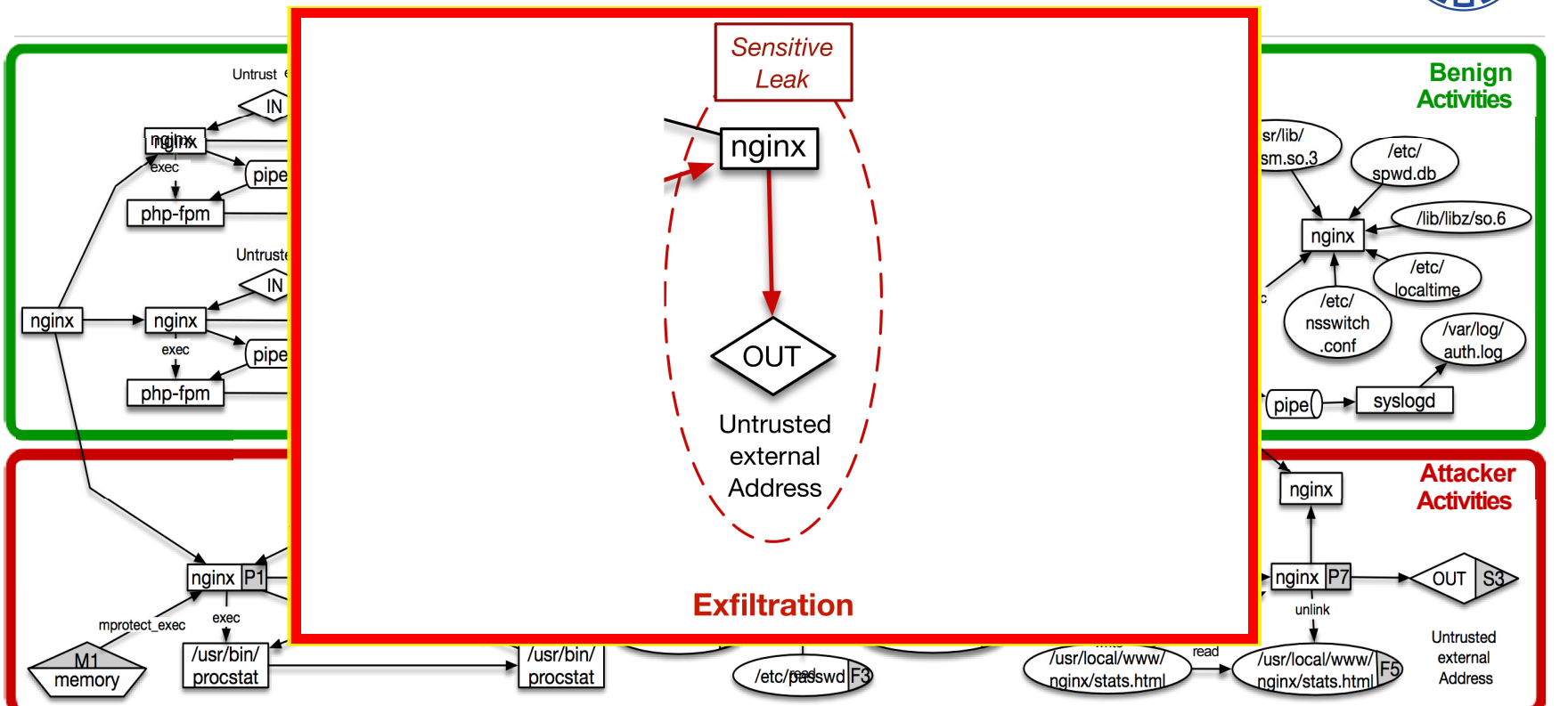
Illustrative Example



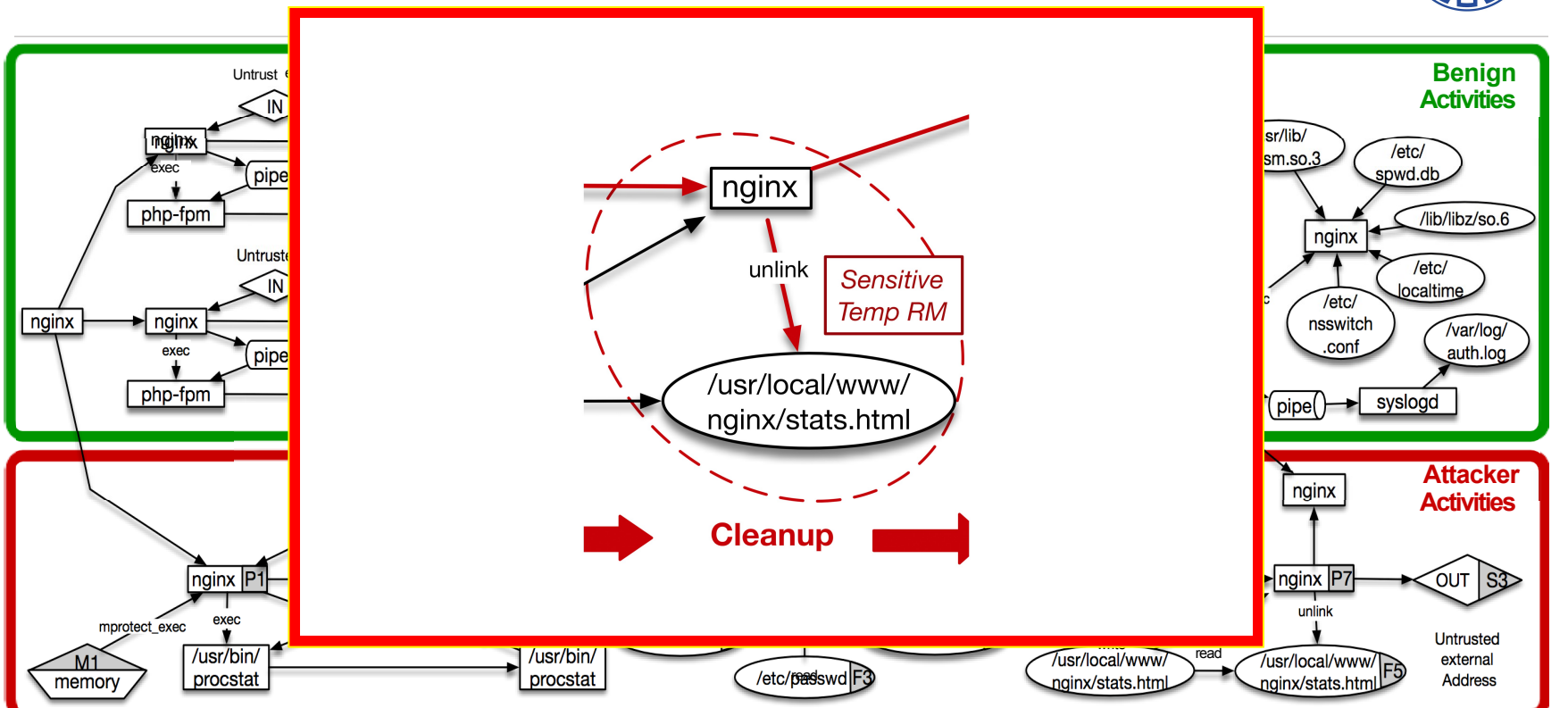
Illustrative Example



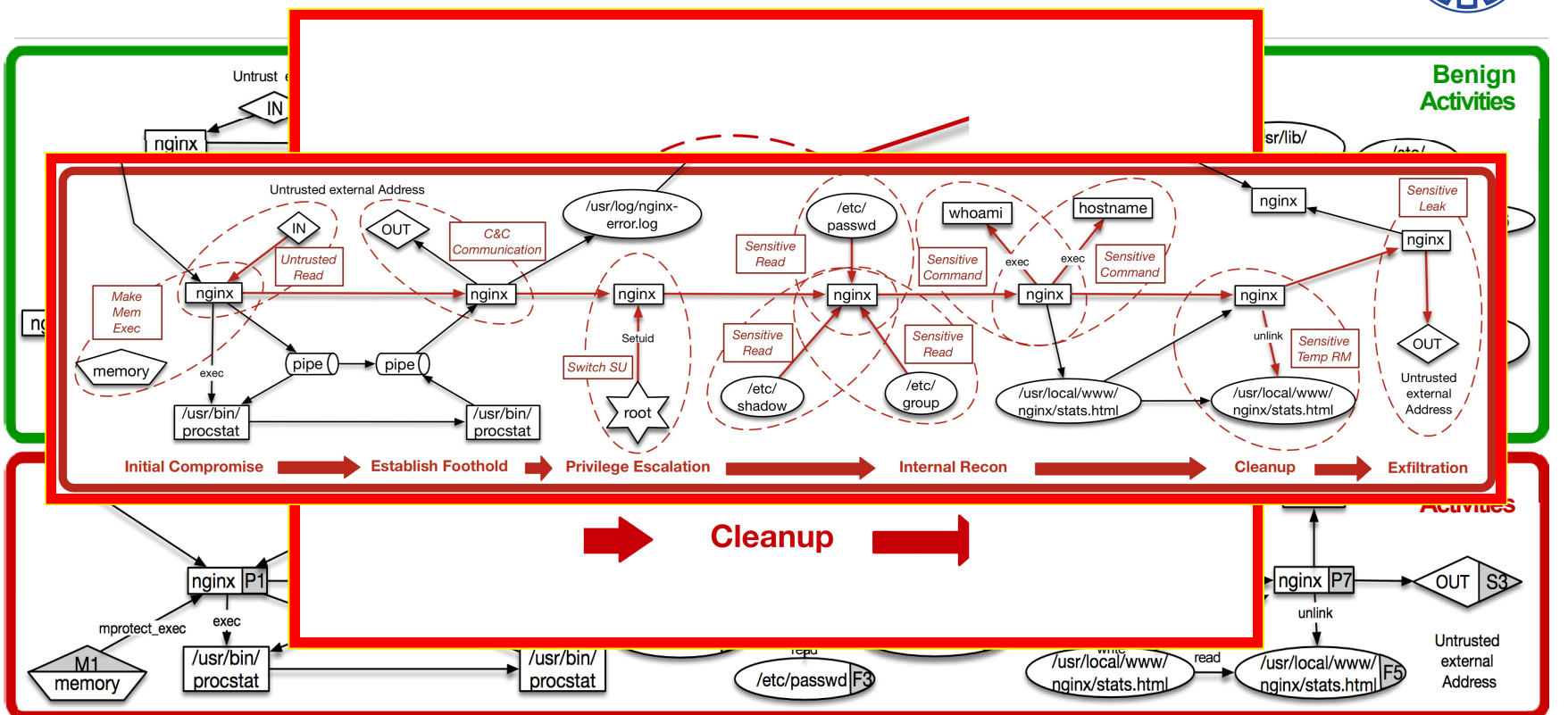
Illustrative Example



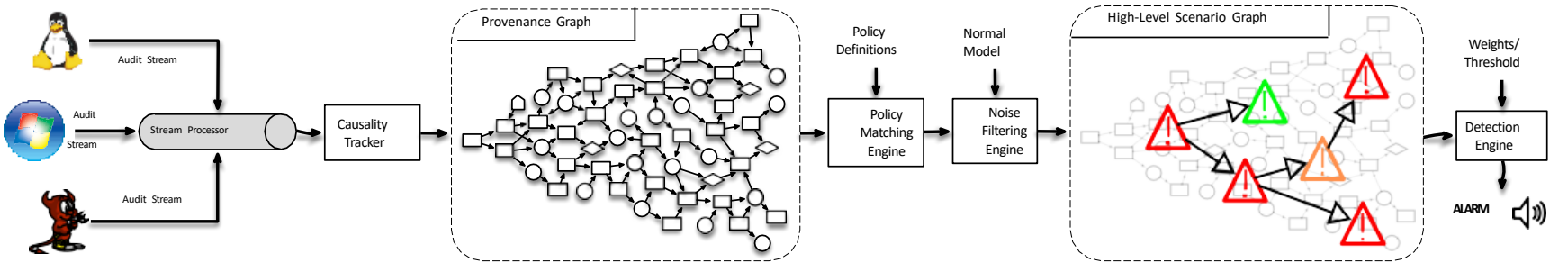
Illustrative Example



Illustrative Example



Holmes Architecture



- Develop TTP specifications over audit logs
- Use specifications to detect TTPs
- Filter noise based on data quantities of benign information flows, measured in bytes transferred
- Construct high-level graph (HSG) that correlates individual alerts/TTPs
- Derive campaign detection signal from graph

Example TTP specifications

APT Stage	TTP	Event Family	Events	Severity	Prerequisites
$Initial\_Compromise(P)$	$Untrusted\_Read(S, P)$	READ	FileIoRead (Windows), read/pread/readv/preadv (Linux,BSD)	L	$S.ip \notin$ {Trusted_IP_Addresses}
	$Make\_Mem\_Exec(P, M)$	MPROTECT	VirtualAlloc (Windows), mprotect (Linux,BSD)	M	\$PROT_EXEC\$ $\in M.flags$ $\wedge \exists\ Untrusted\_Read(?, P') : path\_factor(P', P) <= path\_thres$
$Establish\_Foothold(P)$	$Shell\_Exec(F, P)$	EXEC	ProcessStart (Windows), execve/fexecve (Linux,BSD)	M	$F.path \in$ {Command_Line_Utilities} $\wedge \exists\ Initial\_Compromise(P') : path\_factor(P', P) <= path\_thres$

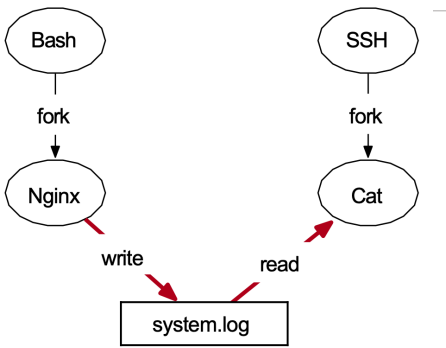
TABLE 4. Example TTPs. In the Severity column, L=Low, M=Moderate, H=High, C=Critical. Entity types are shown by the characters: P=Process, F=File, S=Socket, M=Memory, U=User.

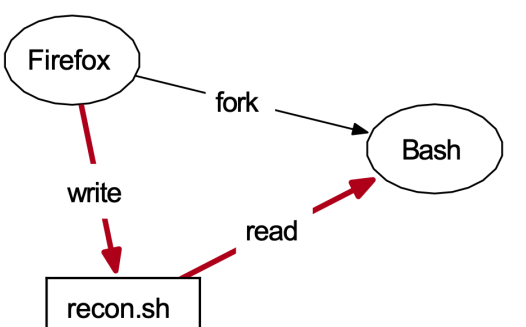
Avoiding spurious dependencies

- Spurious dependencies can result in dependence explosion
- Addressed by asking a key question: what is the influence that attacker had in creating a dependency?
- Key notion Ancestor cover for f : set of all processes that influence a dependency f .

$$\forall p \in f \; \exists a \in AC(f) \;\; a = p \text{ or } a \text{ is an ancestor of } p$$

- Minimal Ancestor cover for f - corresponds to the minimum number of processes attacker should exploit to influence a dependency f.





Avoiding spurious dependencies (Cont.)

$$path\_factor(N_1, N_2) = \min_{\forall f : f.src = N1, f.dst = N2} AC_{min}(f)$$

- *path_factor* value computed incrementally in real-time

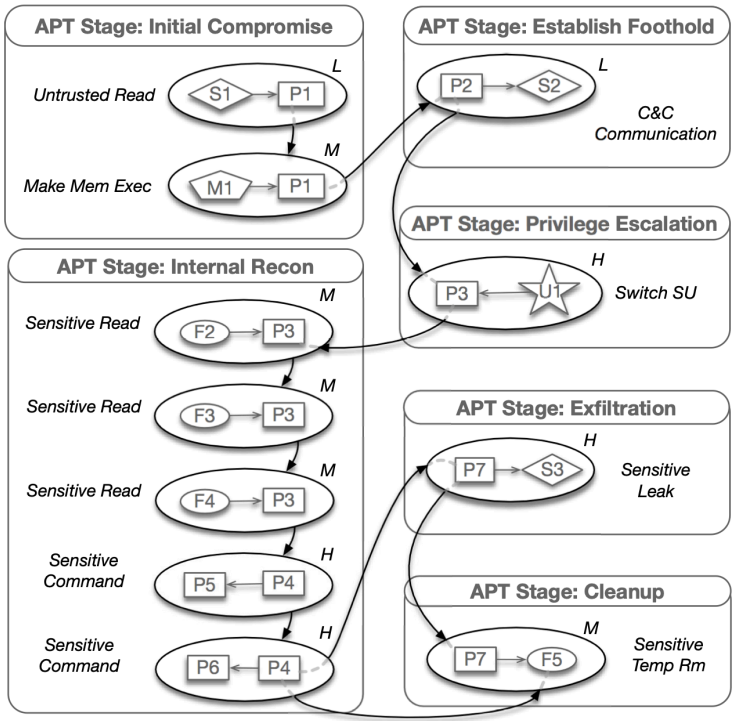
APT Stage	TTP	Event Family	Severity	Prerequisites
$Complete\_Mission(P)$	$Sensitive\_Leak(P, S)$	SEND	H	$S.ip \notin$ {Trusted_IP_Addresses} $\wedge \exists\ Internal\_Reconnaissance(P') : path\_factor(P', P) <= path\_thres$ $\wedge \exists\ Initial\_Compromise(P'') : path\_factor(P'', P) <= path\_thres$

Value of *path_thres* could be set based on the threat an organization is preventing from

- we assume attacker is not willing or capable to compromise more than 3 exploits.

Signal Correlation, HSG, and Threat Tuples

- A TTP is matched and added to the HSG if all its prerequisites are satisfied.
- HSG → Threat Tuple: represents various stages of an APT campaign.
- Each element in tuple takes on severity levels ⟨M, L, H, H, –, H, M⟩
- HSG provides a compact, visual summary of the campaign at any moment.
- cyber-analyst can quickly infer the big picture of the attack (scope and magnitude)



HSG Ranking and Prioritization

- Severity level transformed to a number based on NIST severity score mappings

Qualitative level	Quantitative Range	Rounded up Average Value
Low	0.1 - 3.9	2.0
Medium	4.0 - 6.9	6.0
High	7.0 - 8.9	8.0
Critical	9.0 - 10.0	10.0

- Tuple transformed into numeric value as weighted product

$$\prod_{i=1}^{n} (S_i)^{w_i} \geq \mathcal{T} \qquad w_i = \frac{10 + i}{10}$$

- Alert raised based on threshold learned from benign activity data
  - ⟨C, M, −, H, −, H, M⟩ → ⟨10, 6, 1, 8, 1, 8, 6⟩ → 1163881

Evaluation Datasets

Dataset 1: Using this dataset, we measure the optimal threshold value

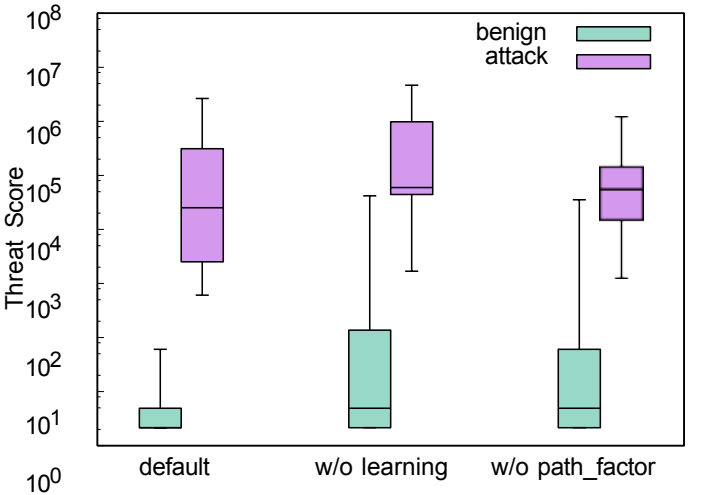
Stream No.	Duration	Platform	Scenario No.	Scenario Name	Atack Surface
1	0d1h17m	Ubuntu 14.04 (64bit)	1	Drive-by Download	Firefox 42.0
2	2d5h8m	Ubuntu 12.04 (64bit)	2	Trojan	Firefox 20.0
3	1d7h25m	Ubuntu 12.04 (64bit)	3	Trojan	Firefox 20.0
4	0d1h39m	Windows 7 Pro (64bit)	4	Spyware	Firefox 44.0
5	5d5h17m	Windows 7 Pro (64bit)	5.1	Eternal Blue	Vulnerable SMB
			5.2	RAT	Firefox 44.0
6	2d5h17m	FreeBSD 11.0 (64bit)	6	Web-Shell	Backdoored Nginx
7	8d7h15m	FreeBSD 11.0 (64bit)	7.1	RAT	Backdoored Nginx
			7.2	Password Hijacking	Backdoored Nginx

Dataset 2: live detection in a setting that we have no prior knowledge of when or how red-team is conducting the attacks.

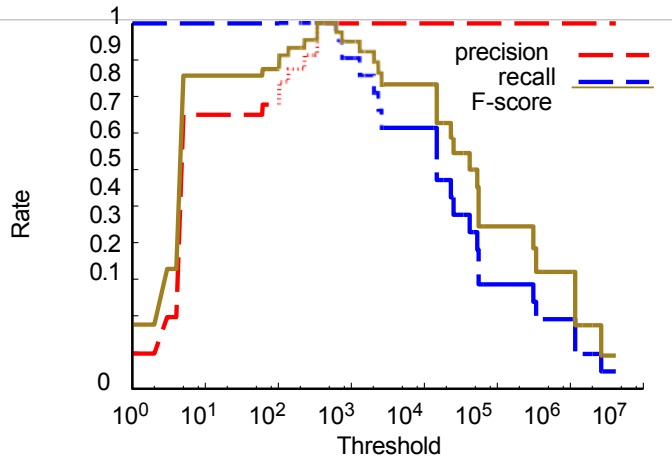
- After this experiment, dataset has been released publicly.

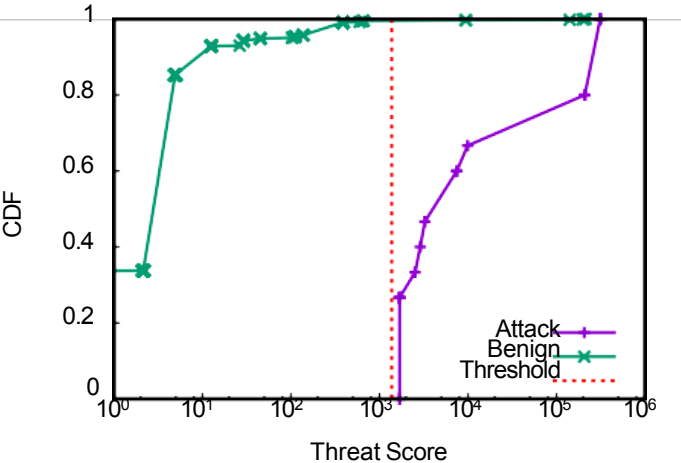
Evaluation

Scenario No.	Threat Tuple	Threat Score	Highest Benign Score in Dataset
1	$\langle C, M, -, H, -, H, M \rangle$	1163881	61
2	$\langle C, M, -, H, -, H, - \rangle$	55342	226
3	$\langle C, M, -, H, -, H, M \rangle$	1163881	338
4	$\langle C, M, -, H, -, -, M \rangle$	41780	5
5.1	$\langle C, L, -, M, -, H, H \rangle$	339504	104
5.2	$\langle C, L, -, -, -, -, M \rangle$	608	
6	$\langle L, L, H, M, -, H, - \rangle$	25162	137
7.1	$\langle C, L, H, H, -, H, M \rangle$	4649220	133
7.2	$\langle M, L, H, H, -, H, M \rangle$	2650614	



Optimal threshold Value &Live Experiment Results





- F-score maximum at [338.25, 608.26] for 6 APT stages
  - Average severity of each APT step = 2.09
- Threshold set for Live experiment (7 APT stages): $2.09^{9.8} = 1378$
- A few false positive: system administrator connecting via SSH

Summary

- Presented a real-time APT detection system that correlates TTPs that might be used to carry out each APT stage.

- visualize high-level APT behavior in real time.
- Dependence explosion mitigation by using the concept of minimum ancestral cover
- Benign system activities pruning based on data quantities in the flow of information
- Experiments show high accuracy and performance for Holmes
- Effectiveness evaluated using a live experiment w/o having prior knowledge of attacks.

Acknowledgments

- [MITRE]https://attack.mitre.org/
- [Holmes] HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows, S. Momeni Milajerdi, R. Gjomemo, B. Eshete,R. Sekar, V. N. Venkatakrishnan, IEEE Symposium on Security and Privacy 2019.