ShadowMove: A Stealthy Lateral Movement Strategy

Jinpeng Wei

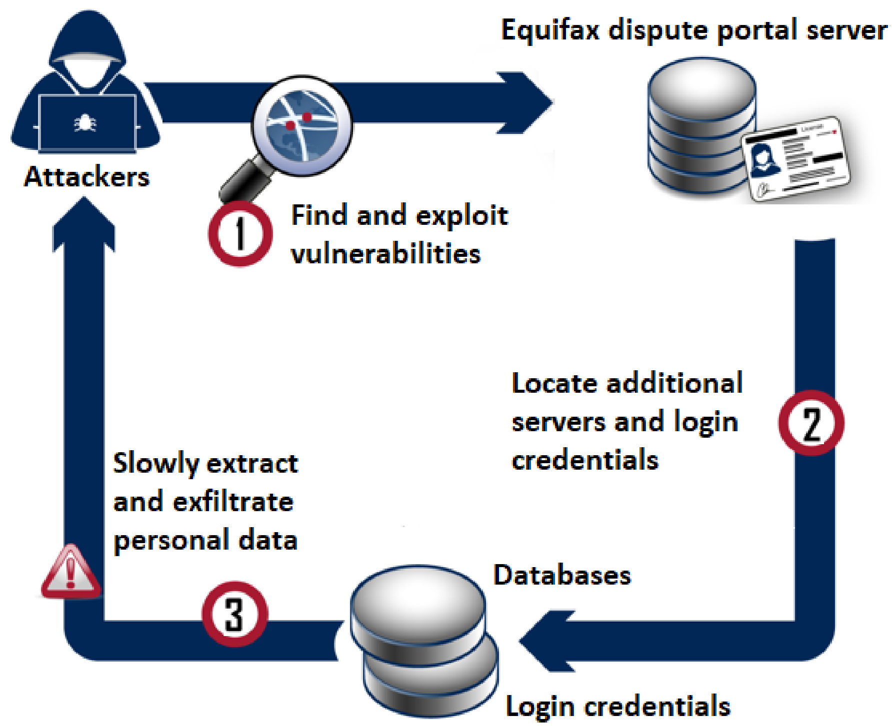
Associate Professor
Department of Software and Information Systems
College of Computing and Informatics
University of North Carolina at Charlotte
https://webpages.charlotte.edu/jwei8/

CAE Forum
April 6, 2022

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

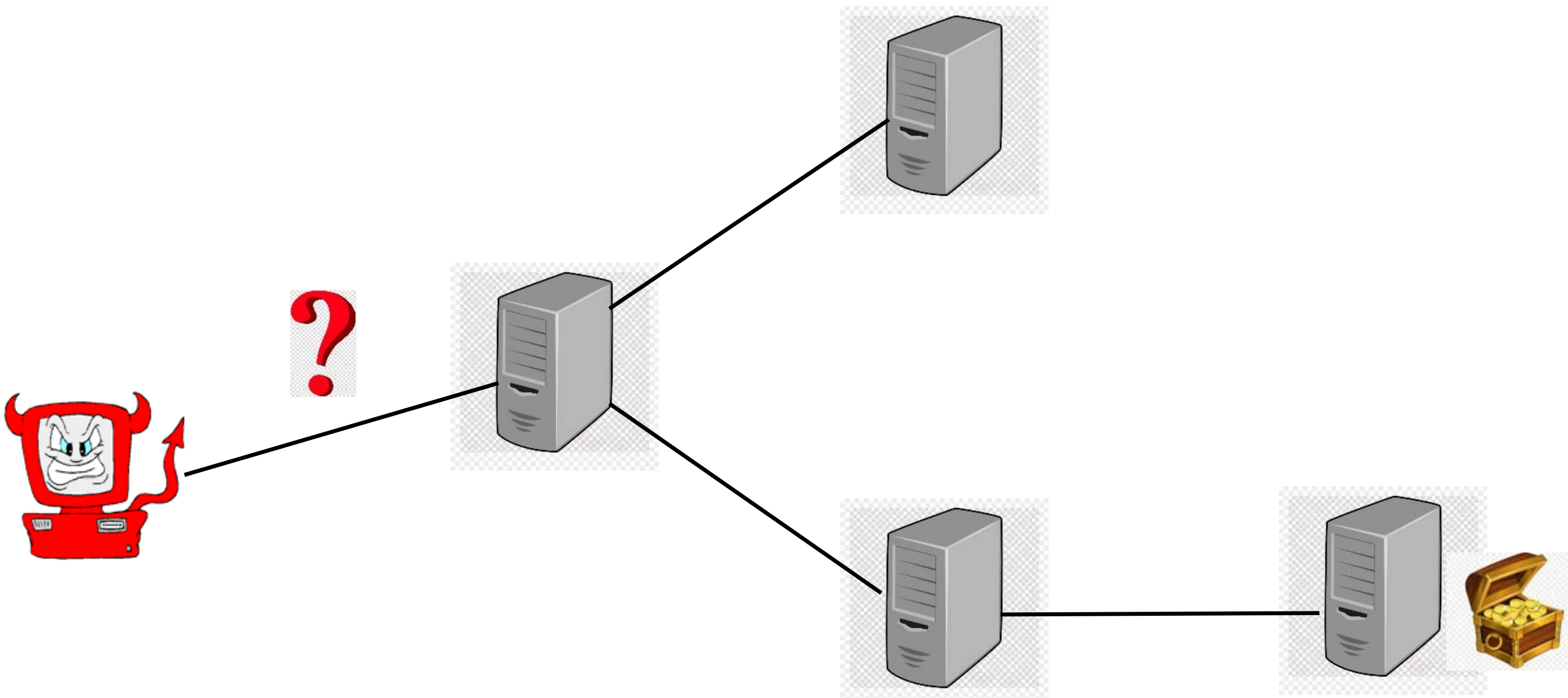
Advanced Persistent Threats (APTs) are Extremely Harmful

- Real world example: Equifax breach



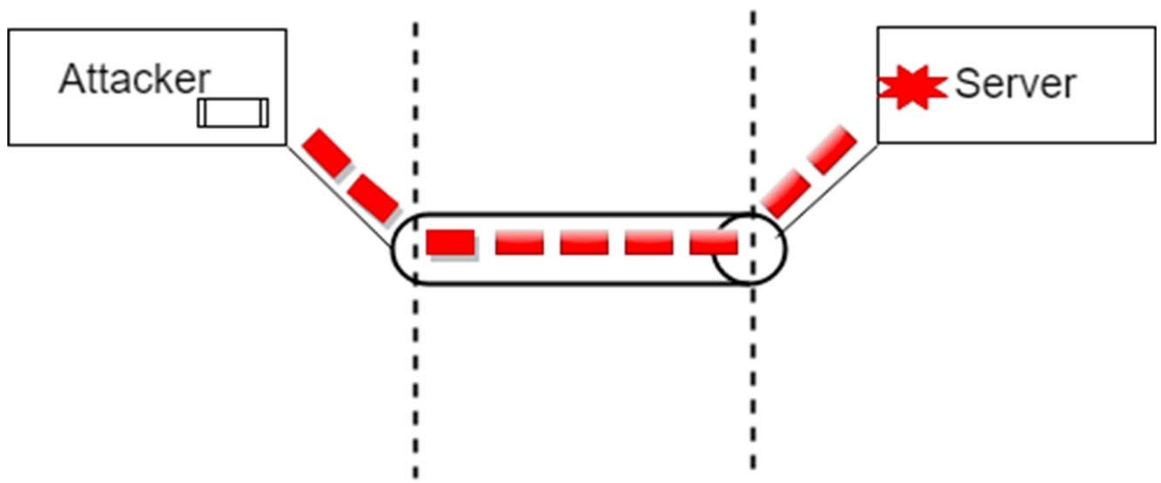
- Features of lateral movement in APT attacks
  - Assume that a foothold within the target network is already established
  - Use the compromised systems as stepping stones to reach critical assets

How can an APT attacker move laterally?



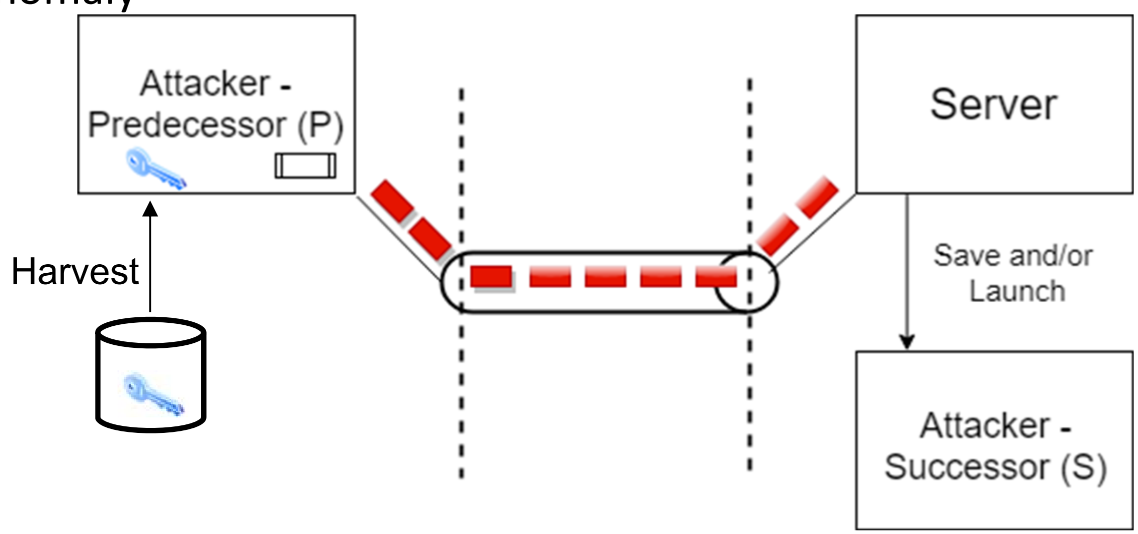
Existing Lateral Movement Technique 1: Vulnerability Exploitation

- Idea: exploit vulnerabilities in network services (e.g., WannaCry exploited an SMB vulnerability)
- Limitations
  - Server must have an unpatched vulnerability
  - Exploitation easy to detect (intrusive to the server, by violating its integrity)



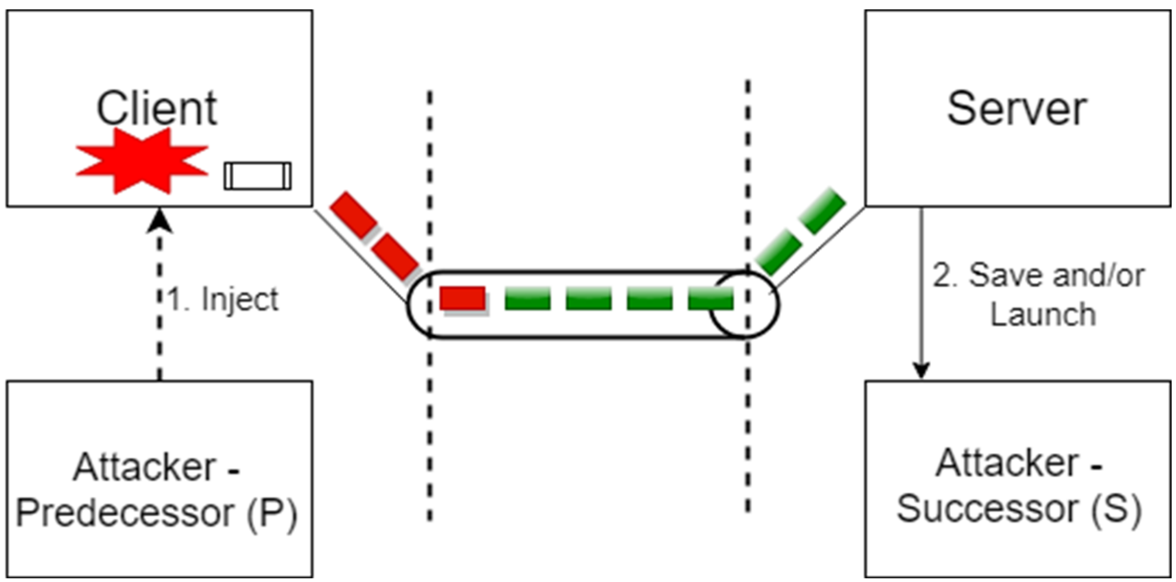
Existing Lateral Movement Technique 2: Credential Abuse

- Idea: harvest and abuse user credentials (e.g., passwords by Equifax breach)
- Limitations
  - Credentials are not easy to get: they may not be saved, they are often encrypted, or key loggers are easy to detect
  - Credentials alone may not be enough (e.g., two-factor authentication)
  - Abusing credentials requires new network connections, which can be detected as anomaly



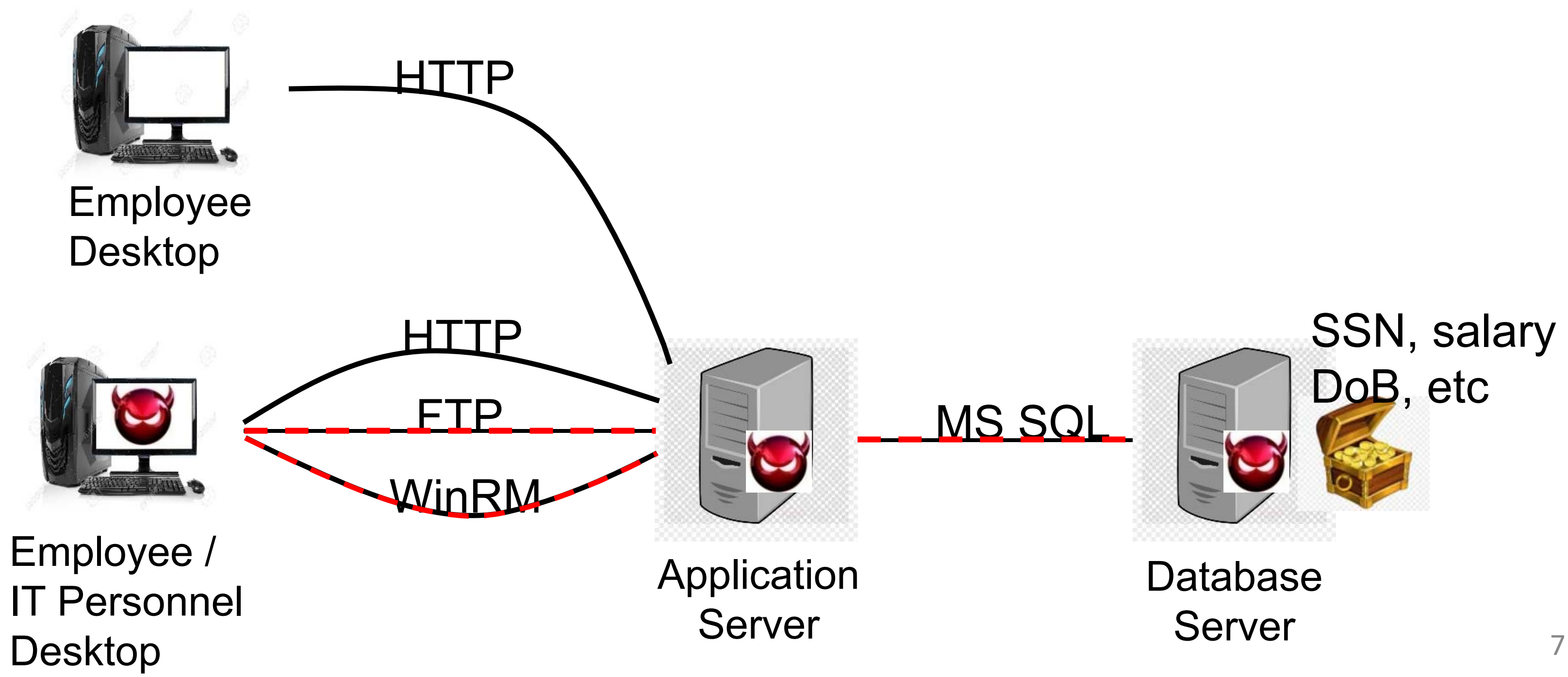
Existing Lateral Movement Technique 3: Process Injection

- Idea: Inject application- and protocol-specific code into legitimate clients to reuse its connections (e.g., SSH-Jack)
- Limitations
  - Complex and hard to implement (application specific)
  - Intrusive to the client (i.e., by violating its integrity) and can be detected by existing defensive solutions (e.g., Windows Defender ATP)



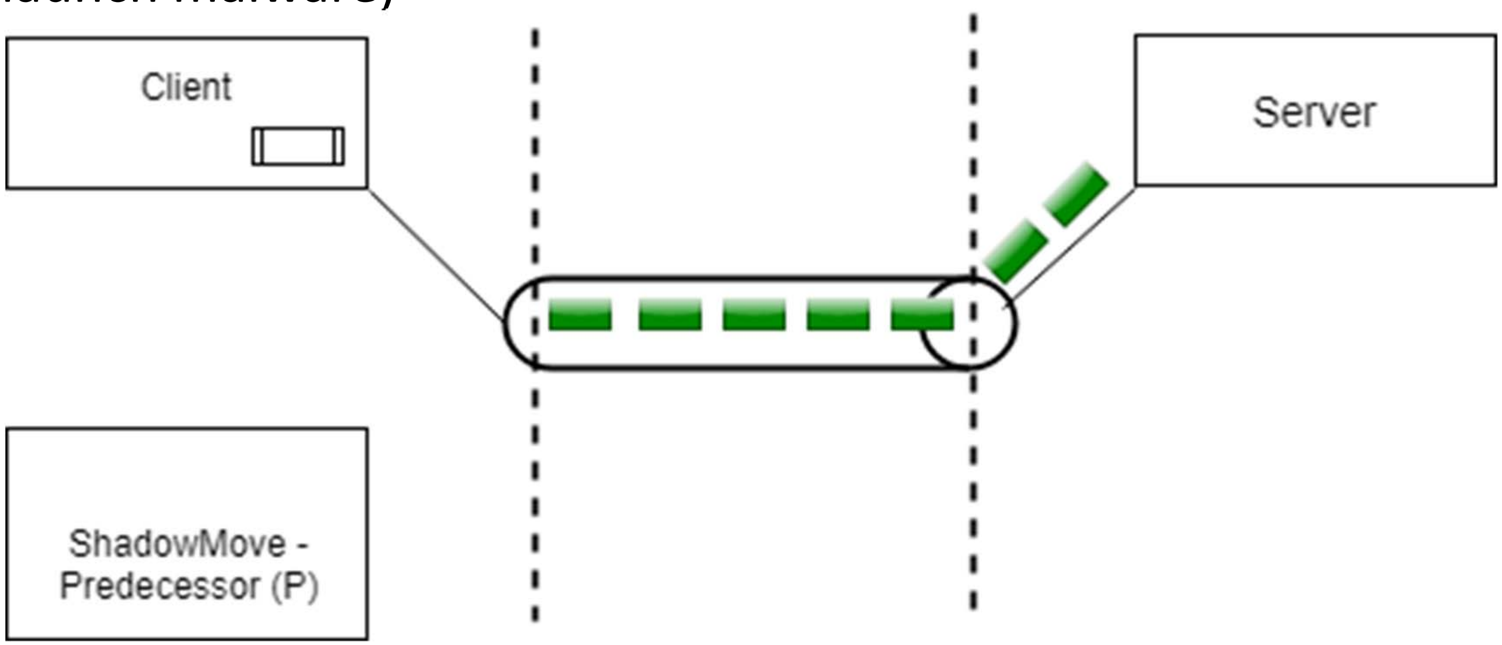
ShadowMove: a Novel Attack Technique

- The idea: silently reuse existing and legitimate network connections to move laterally towards valuable targets in a compromised enterprise network
- Example: Employee Self-Service Application (e.g., using FTP to upload malware and WinRM to launch malware)



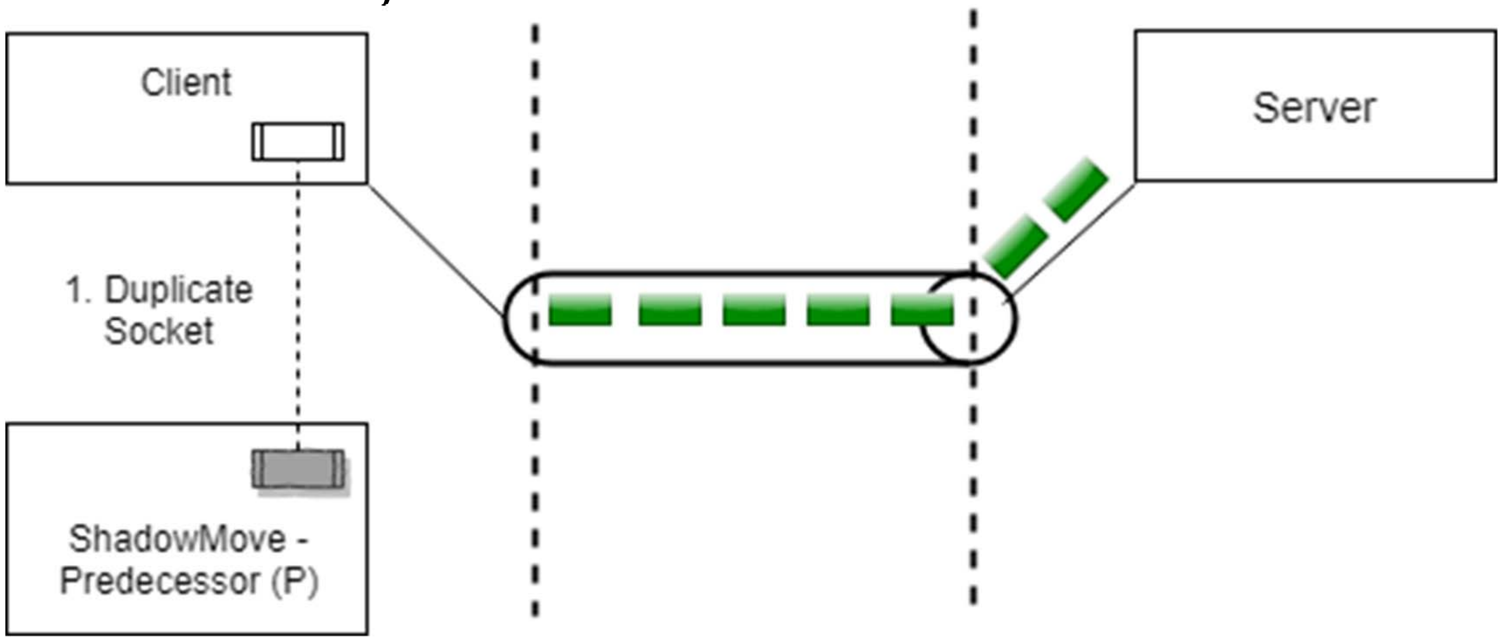
ShadowMove: a Novel Attack Technique

- The idea: silently reuse existing and legitimate network connections to move laterally towards valuable targets in a compromised enterprise network (e.g., FTP to upload malware, and WinRM to launch malware)



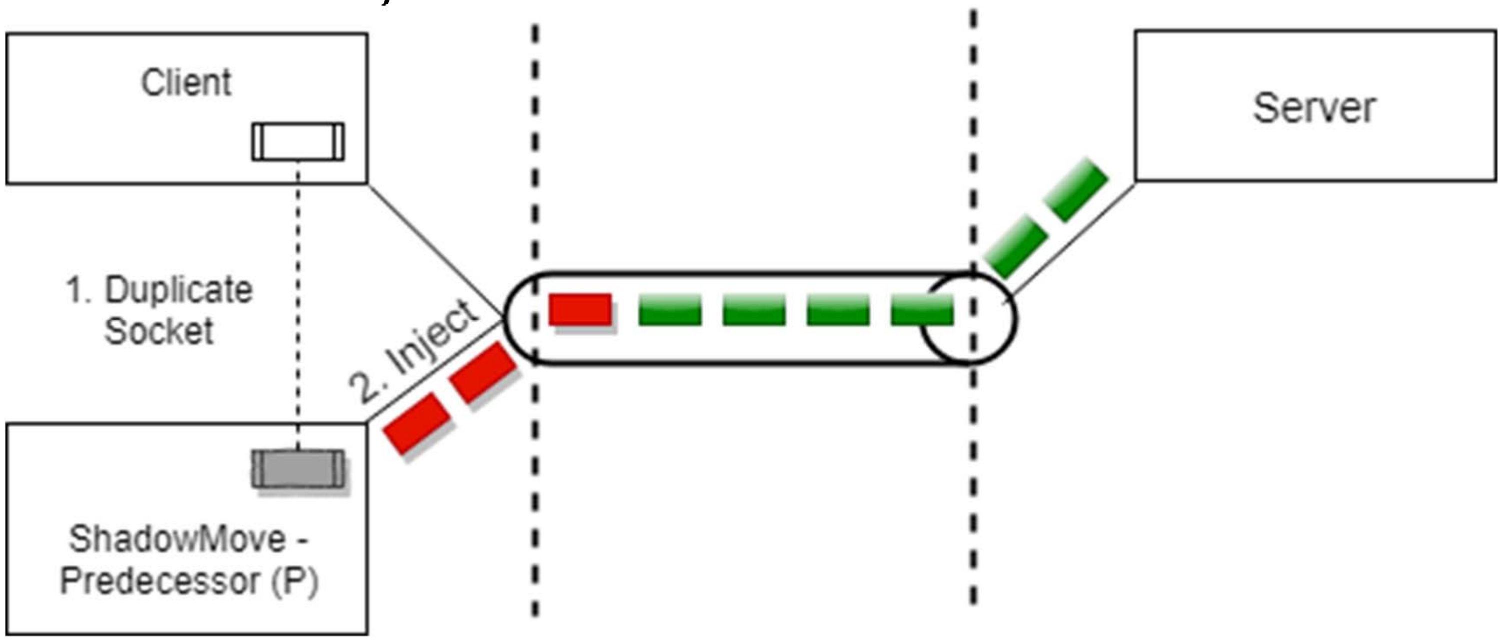
ShadowMove: a Novel Attack Technique

- The idea: silently reuse existing and legitimate network connections to move laterally towards valuable targets in a compromised enterprise network (e.g., FTP to upload malware, and WinRM to launch malware)



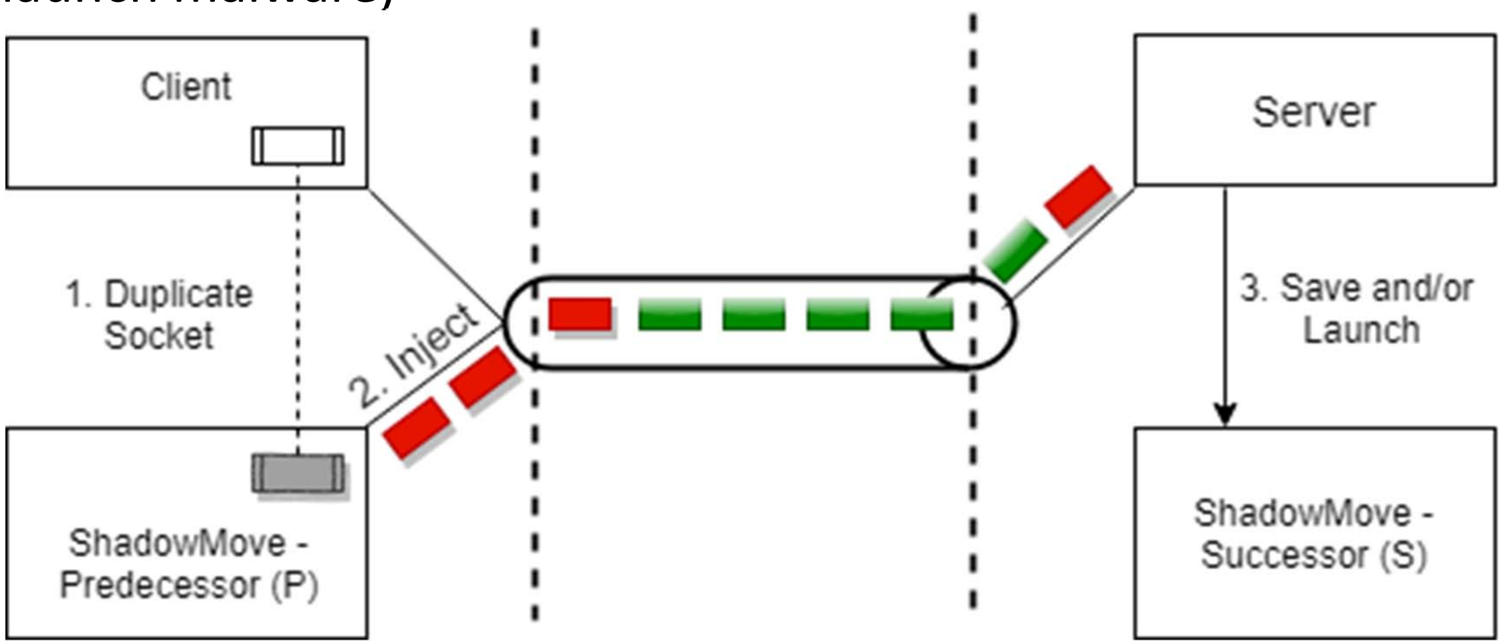
ShadowMove: a Novel Attack Technique

- The idea: silently reuse existing and legitimate network connections to move laterally towards valuable targets in a compromised enterprise network (e.g., FTP to upload malware, and WinRM to launch malware)

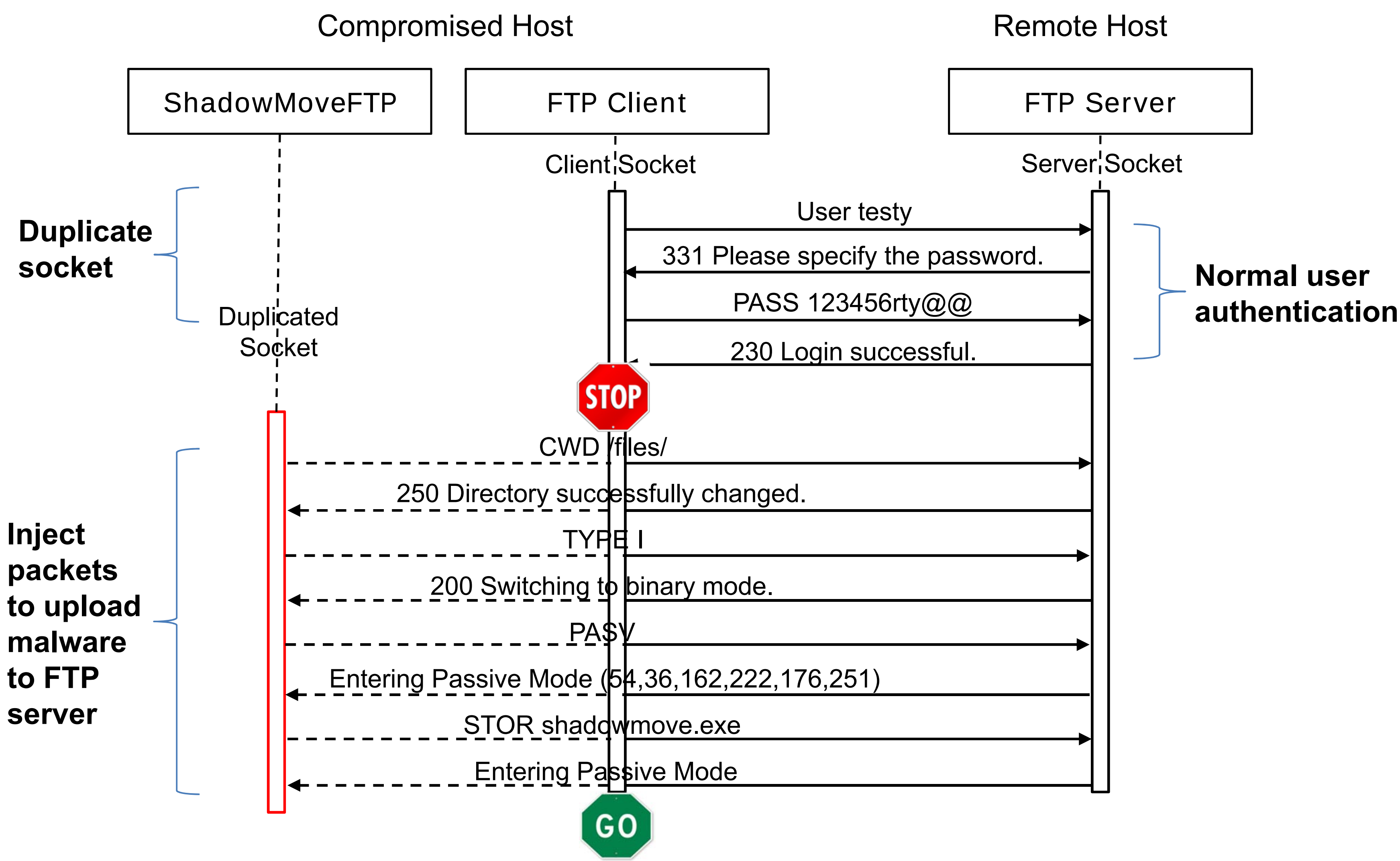


ShadowMove: a Novel Attack Technique

- The idea: silently reuse existing and legitimate network connections to move laterally towards valuable targets in a compromised enterprise network (e.g., FTP to upload malware, and WinRM to launch malware)

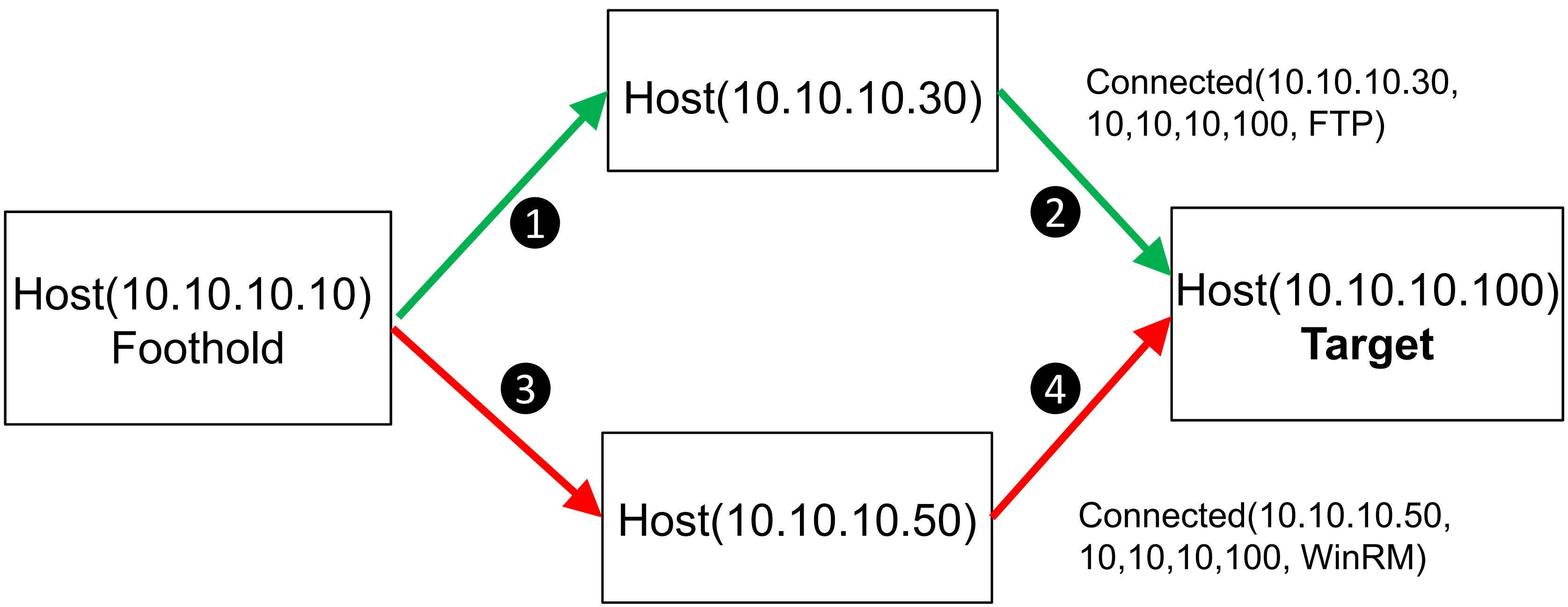


Case Study: Single Hop ShadowMove Over FTP



ShadowMove among Network Nodes

- From foothold to valuable target buried in the network
- Based on a global network view learned over time



ShadowMove Makes APT Attacks Easier

- Traditional lateral movement techniques are **application-specific**
  - Vulnerability exploitation: case-by-case, server-specific, not scalable
  - Credential abuse: server-specific, not scalable
  - Process injection: case-by-case, not scalable

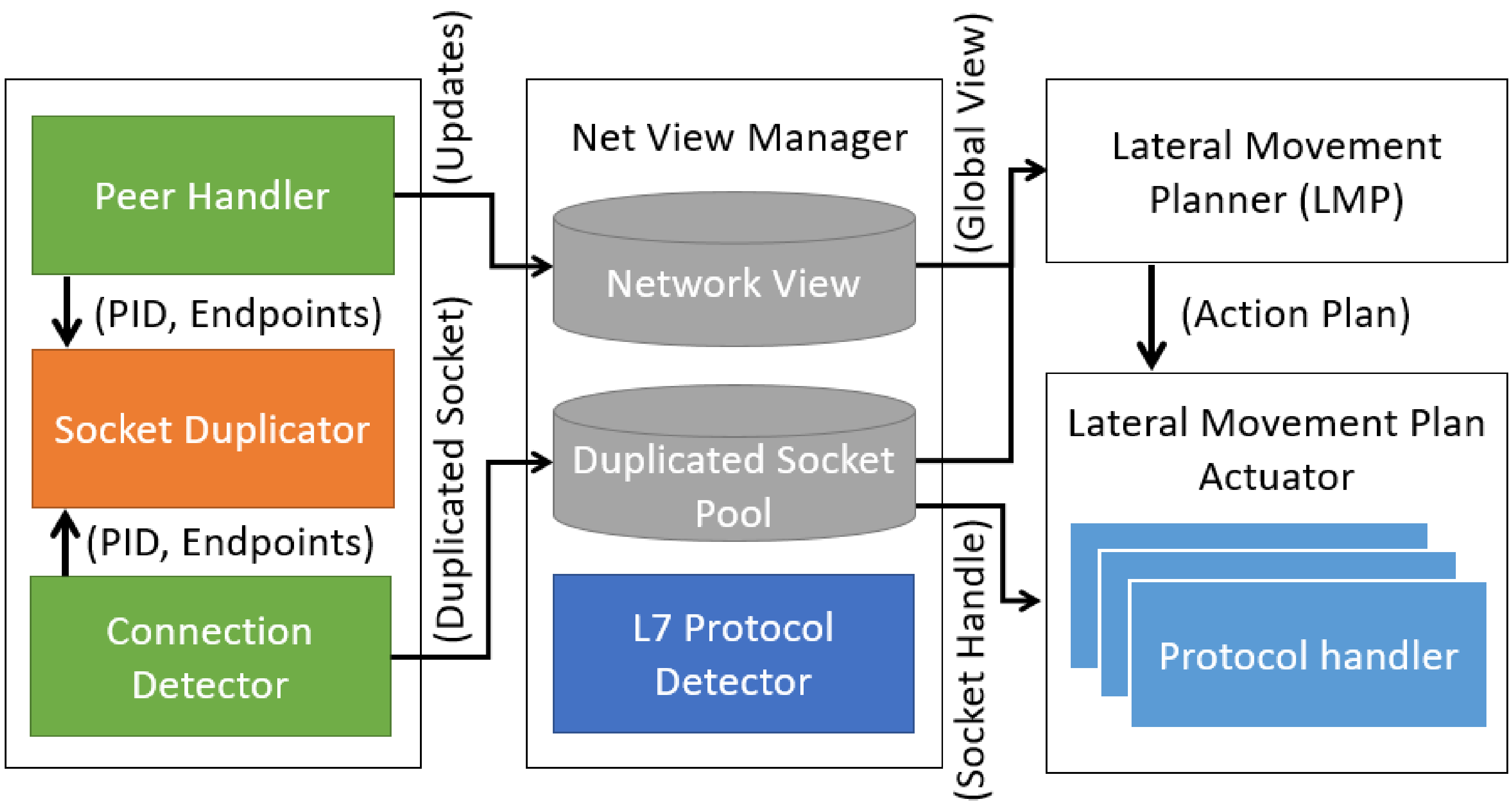
- ShadowMove is general and scalable: works for any server and any client, targeting protocols instead of applications

- ShadowMove has plenty of opportunities to thrive in enterprise computing environments:
  - A large number of protocols, most having public specifications
    - E.g., FTP, WinRM, Microsoft SQL, HTTP, AJP, MySQL SQL, MQTT, etc
  - Widespread automation (e.g., Passwordless SSH Login)
  - Weaker protection inside enterprise networks: traffic often not encrypted

Threat Model

- Attackers have established a foothold on a victim system under a normal user's privilege
  - Easy to satisfy: given the prevalence of malware infection, caused by spearphishing and drive-by downloads
- The attackers run malware to automatically move towards the critical asset(s)
- The victim process whose TCP connection is going to be hijacked is not aware of the malware process

ShadowMove Architecture



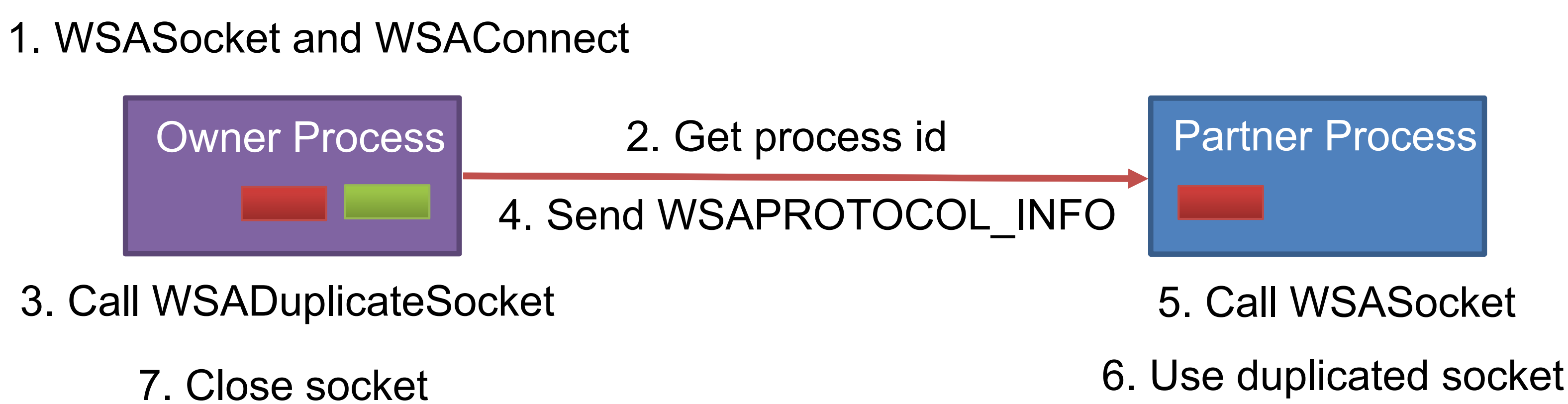
Connection Detector

Detects newly created sockets suitable for duplication

- Periodically gets list of TCP connections
  - E.g. by calling GetTcpTable2 and GetTcp6Table2
- Identifies new connections
- Filters out the ones owned by a process that cannot be accessed
- Calls socket duplicator to duplicate the sockets for new connections

Conventional Socket Duplication

- Official socket duplication requires cooperation of socket owner
- Example: Windows socket duplication



ShadowMove Socket Duplicator

- ShadowMove invokes Windows APIs in an unconventional way. Therefore, it does not require cooperation from the socket owner process.

Step	Description	kernel/ntdll functions
1	Open the owner process with PROCESS_DUP_HANDLE	OpenProcess(PROCESS_DUP_HANDLE, , pid)
2	Foreach handle with type 0x24 (file)	NtQuerySystemInformation(SystemHandleInformation, ...)
3	Duplicate the handle	NtDuplicateObject
4	Retrieve its names	NtQueryObject(ObjectNameInformation)
5	Skip if the name is not \device\afd	
6	Obtain remote IP and remote port number	getpeername(handle, ...)
7	Skip if remote IP and port do not match the input parameters	
8	Call WSADuplicateSocketW to get a special WSAPROTOCOL_INFO structure	WSADuplicateSocketW(handle, ...)
9	Create a duplicate socket	WSASocketW(WSAPROTOCOL_INFO, ...)
10	Use the socket	recv(), send()

Peer Handler

Constructs a global view of the compromised network by synchronizing its current view with neighboring ShadowMove instances

- Receives network views from neighboring nodes
  - Peeks from duplicated sockets
  - waits for synchronization signal
- Sends synchronization signal periodically to its predecessor/successor nodes
  - Sends its own network views

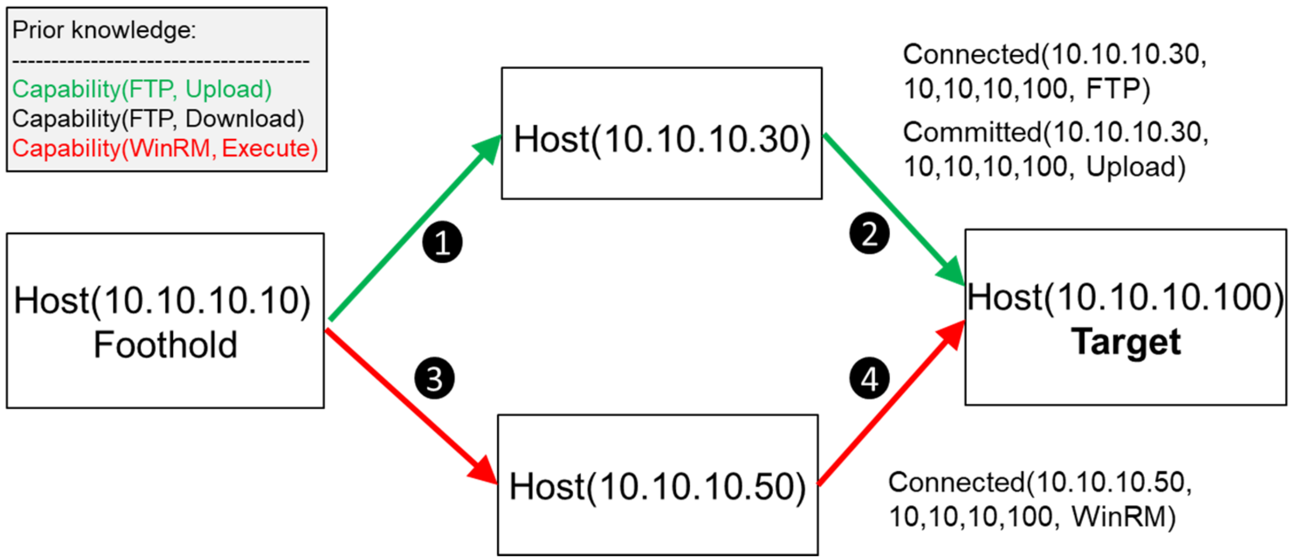
Lateral Movement Planner

- Formulates the next lateral movement action plan based on
  - Current network view
  - History of action plans performed by all ShadowMove instances
- Optimizes for effectiveness and stealth
- Logic programming based

An action plan describes the action that must be performed on a specific end point

Lateral Movement Planner

Logic Programming predicates



Logic Programming rules

commitExecuteOperation(X, Y) :- connected(X, Y, Z), capability(Z, execute), origin(I), remoteOperation(I, Y, upload, _R), committed(_K, Y, upload).

remoteOperation(X, Y, Action, Route):- connected(X, Z, Service), capability(Service, Action), remoteOperation(Z, Y, Action, R), Route=[X| R].

Plan

Use WinRM connection (4) to launch malware on the Target

Lateral Movement Plan Actuator

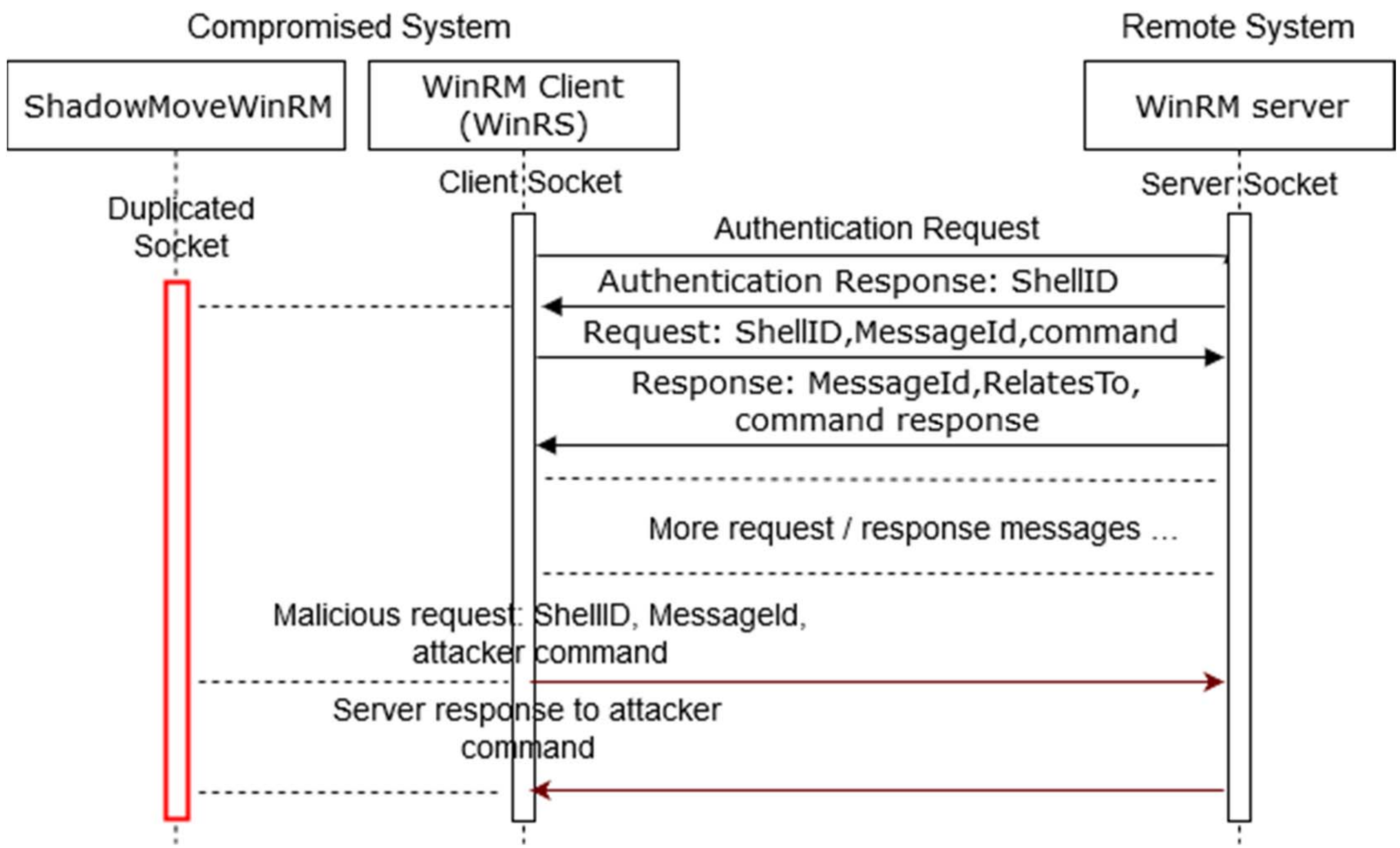
Creates protocol-specific queries to carry out lateral movement plans

- Contains a set of Protocol Handlers
  - Application protocol specific
    - FTP, TDS (MS SQL), and WinRM
  - Performs different operations
    - Upload, Download, or Execute

ShadowMove Implementation

- We implement a prototype of the ShadowMove design on Windows in 2,501 lines of C/C++ code
  - We also have a simpler prototype for Ubuntu Linux
- The lateral movement planner is based on SWI-Prolog

ShadowMove PoC Leveraging WinRM (Windows Remote Management)



Why are ShadowMove Attacks Possible?

- The conflicting requirements between process isolation and resource sharing in commodity OS
  - allows the attack process to duplicate (share) sockets belonging to legitimate client processes.
- A lack of built-in message origin integrity validation in many networking protocols
  - allows malicious packets in existing connections that cannot be differentiated from legitimate packets.

Evaluation of the Stealthiness

- Not detected by off-the-shelf defense solutions

Type	Name	Version	Update time	FTP/MSSql/WinRM
AV	McAfee	16.0	3 Feb 2019	N/N/N
AV	Norton	22.16.2.22	3 Feb 2019	N/N/N
AV	Webroot	9.0.24.37	3 Feb 2019	N/N/N
AV	Bitdefender	6.6.7.106	3 Feb 2019	N/N/N
AV	Windows Defender	4.18.1901.7	3 Feb 2019	N/N/N
NIDS	Snort (Windows and Linux)	2.9.12	7 Feb 2019	N/N/N
HIDS	OSSEC (Linux)	3.4.0	12 Oct 2019	N/--/--
HIDS	Osquery (Linux)	4.0.2	24 Oct 2019	N/--/--
HIDS	Wazuh (Linux)	3.10.2	24 Oct 2019	N/--/--
EDR	Cisco AMP	6.1.5.10729	14 Jun 2018	N/N/N
EDR	CrowdStrike Falcon Prevent	4.20.8305.0	11 Feb 2019	N/N/N

Limitations of the Current ShadowMove Prototype

- It cannot hijack connections for which user-level encryption is applied to the payload
- It may not be able to get information such as the shellID in WinRM attack from the receiving buffer if it misses the authentication phase
- Our design of ShadowMove on Linux relies on a small amount of code injection

Conclusion

- We present the ShadowMove strategy that allows APT attackers to make stealthy lateral movements within an enterprise network
- ShadowMove leverages existing benign network connections and does not require any elevated privilege, new connections, extra authentication, or process injection
- We developed a prototype of ShadowMove for modern versions of Windows and Linux OSes, which successfully abuses three common enterprise protocols (i.e., FTP, Microsoft SQL, and WinRM)
- We also experimentally confirm that our prototype implementation is undetectable by state-of-the-art antivirus products, IDSes (such as Snort), and Endpoint Detection and Response systems

Acknowledgement

Bei-Tseng Chu
UNC Charlotte

Qingyang Wang
LSU

Amir Niakanlahiji
Microsoft

Md Rabbi Alam
UNC Charlotte

Questions?