Operating Systems and Program Security

Kc Udonsi

Application Security & Threat Modelling

Common Threats Against Software

- ➡ Presence of security bugs "Vulnerabilities"
- ➡ Unauthorized modification e.g Backdoors
- ➡ Supply chain bugs - Vulnerabilities in dependencies and/or tooling, partners

Why do Vulnerabilities exist ?

- ➡ Fundamental oversights in software design. Designed to do the wrong thing a.k.a Design Flaws
- ➡ Implementation flaws/bugs relevant to security a.k.a Technical Flaws
- ➡ Faulty inter-operation with executing environment a.k.a Operational Flaws
- ◉ **Arbitrarily trusting input data, misplaced trust**

Threat Modelling

- ➡ Description of system
- ➡ Potential threats to the system (threats against CIA)
- ➡ Actions that can be taken to mitigate each threat
- ➡ Validation of model
- ➡ Threat Modelling Manifesto: https://www.threatmodelingmanifesto.org/
- ➡ Think about "abuse cases" and what can be done to mitigate those

Secure Programming

- ➡ Familiarity with relevant vulnerability classes
- ➡ Modularity - separate modules for separate functionalities
- ➡ Sanitize, validate, restrict input data even between modules or components (mutual suspicion)
- ➡ Be "fault tolerant" by having a consistent policy to handle failure
- ➡ Use reputable, security conscious and well maintained libraries
- ➡ Adopt good programming practices, be security aware

Software Security Assessment

- Manual, guided or automated audit and security testing
- Security test cases may validate threat mitigation strategies
- Internal or external auditors methodologically review code for design, implementation or operational flaws
    - Vulnerability Rewards Program, Bug Bounties etc
- Fuzz testing can be combined with manual audits to discover vulnerable code paths
- Can be carried out at various stages of the SDLC

Secure Software Development Life Cycle

- ➡ Description of subject
- ➡ Potential threats to the system
- ➡ Actions that can be taken to mitigate each threat
- ➡ Validation of model
- ➡ Continuous security testing throughout the SDLC "DevSec Ops"
- ➡ Think about "abuse cases" and what can be done to mitigate those

Formal Methods of Verification

Mathematical description of the problem

Refinement steps

Proof of correctness

Executable code
or hardware design

Formal Methods of Verification

➡ Examples:

Hardware design (VHDL, Verilog)

✓ Used by semi-conductor companies such as Intel

Critical embedded software (B/Z, Lustre/Esterel)

✓ Urban Transportation
(METEOR Metro Line 14 in Paris by Alstom)

✓ Rail transportation (Eurostar)

✓ Aeronautic (Airbus, Eurocopter, Dassault)

✓ Nuclear plants (Schneider Electric)

Pros and cons of using formal methods

- ✓ Nothing better than a mathematical proof
- ➡ A code "proven safe" is safe
- ◉ Development is time and effort (and so money) consuming
- ➡ Should be motivated by the risk analysis
- ◉ Do not prevent from specification bugs
- ➡ Example of network protocols

Operating System Security

Exploit mitigation, Endpoint Detection and Response (EDRs), Security Policies

Exploit Mitigation

Exploit Mitigation Contd.

- Fortify Source Functions
- Stack Canaries
- Data Execution Prevention / Non-Executable Stack
- Address Space Layout Randomization (ASLR)

Exploit Mitigation Contd.

- ➡ Position Independent Executables
- ➡ Control Flow Guard
- ➡ Application sandboxing
- ➡ Non-exhaustive. Often implemented at OS or Compiler

Fortify Source Functions

➡ GCC macro FORTIFY_SOURCE provides buffer overflow checks for unsafe C libraries

```
memcpy, mempcpy, memmove, memset, strcpy,
stpcpy, strncpy, strcat, strncat, sprintf,
vsprintf, snprintf, vsnprintf, gets
```

Checks are performed

- some at compile time (compiler warnings)
- other at run time (code dynamically added to binary)

Canaries

- The compiler modifies every function's prologue and epilogue regions to place and check a value (a.k.a a canary) on the stack
- When a buffer overflows, the canary is overwritten. The programs detects it before the function returns and an exception is raised
- Different types:
  - random canaries
  - xor canaries
- Disabling Canary protection on Linux
  `$ gcc ... -fno-stack-protector`
- Bypassing canary protection : *Structured Exception Handling (SEH)* exploit overwrite the existing exception handler structure in the stack to point to your own code

DEP/NX - Non Executable Stack

- The program marks important structures in memory as non-executable
- The program generates an hardware-level exception if you try to execute those memory regions
- This makes normal stack buffer overflows where you set `eip` to `esp+offset` and immediately run your shellcode impossible
- Disabling NX protection on Linux
  `$ gcc ...-z execstack`
- Bypassing NX protection : *Return-to-lib-c* exploit
  return to a subroutine of the lib C that is already present in the process' executable memory

ASLR - Address Space Layout Randomization

- The OS randomize the location (random offset) where the standard libraries and other elements are stored in memory
- Harder for the attacker to guess the address of a lib-c subroutine
- Disabling ASLR protection on Linux
  ```
  $ sysctl kernel.randomize_va_space=0
  ```
- Bypassing ASLR protection : Brute-force attack to guess the ASLR offset
- Bypassing ASLR protection : *Return-Oriented-Programming (ROP)* exploit use instruction pieces of the existing program (called "gadgets") and chain them together to weave the exploit

PIC/PIE - Position Independent Code/Executables

- **Without PIC/PIE**
  code is compiled with absolute addresses and must be loaded at a specific location to function correctly

- **With PIC/PIE**
  code is compiled with relative addressing that are resolved dynamically when executed by calling a function to obtain the return value on stack

Confined execution environment - Sandbox

A sandbox is tightly-controlled set of resources for untrusted programs to run in

➡ Sandboxing servers - virtual machines

➡ Sandboxing programs

- Chroot, Seccomp, AppArmor in Linux
- Sandbox in MacOS
- Application Guard Windows
- Windows Sandbox

➡ Sandboxing applets - Java and Flash in web browsers

Security Policies

Baselining System Security

- ➡ OSes strive for secure out-of-the-box
- ➡ Granular controls may be required to customize security posture
- ➡ Often pushed down as configurations or profiles in enterprise environment
- ➡ May include firewall settings, password strength requirements, application installations, removal drive controls, suspicious site access, file download policies etc.

Vulnerability Management

To Patch or Not to Patch …

- ➡ Patches often need to be validated
- ➡ Risk-based discovery, prioritization and remediation

Securing the Kernel

Kernel Patch and Exploit Mitigations

- ➡ Kernel Self-Protection (Linux)
- ➡ Kernel Patch Guard / Patch Protection (KPP) (Windows)
- ➡ Kernel Data Protection (Windows)
- ➡ System Coprocessor / Kernel Integrity Protection (MacOS)
- ➡ Pointer Authentication Codes (MacOS)
- ➡ Code integrity and signing
- ➡ Non-exhaustive. Often implemented at OS or hypervisor level (Virtualization Based Security)

Endpoint Detection and Response

Endpoint Protection

➡ Historic anti-virus - signature based detection

➡ Heuristics and behavioural based detection

➡ Implemented as an extension to the kernel often with user-space components

➡ Passive or Active mode, event logging and streaming

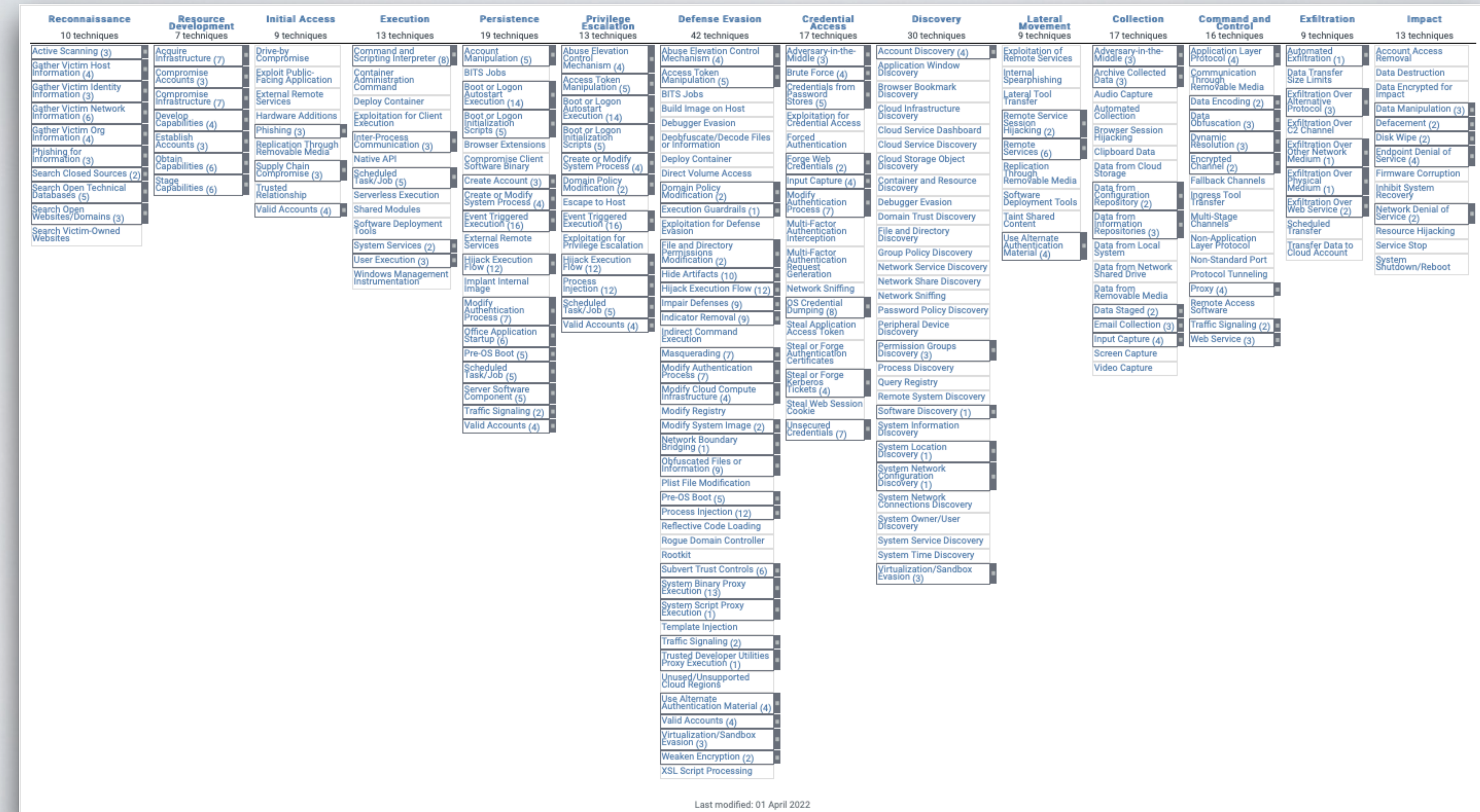
➡ Often featuring a cloud component for incident investigation and security overview

➡ Still software hence can be contain vulnerabilities

Endpoint Protection

➡ Mitre Attack Matrix

➡



Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
10 techniques	7 techniques	9 techniques	13 techniques	19 techniques	13 techniques	42 techniques	17 techniques	30 techniques	9 techniques	17 techniques	16 techniques	9 techniques	13 techniques
Active Scanning (3)	Acquire Infrastructure (7)	Drive-by Compromise	Command and Scripting Interpreter (8)	Account Manipulation (5)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Adversary-in-the-Middle (3)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (3)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Compromise Accounts (3)	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (3)	Compromise Infrastructure (7)	External Remote Services	Deploy Container	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	BITS Jobs	Credentials from Password Stores (5)	Browser Bookmark Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Gather Victim Network Information (6)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Gather Victim Org Information (4)	Establish Accounts (3)	Phishing (3)	Inter-Process Communication (3)	Browser Extensions	Create or Modify System Process (4)	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (6)	Browser Session Hijacking	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Compromise Client Software Binary	Domain Policy Modification (2)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Search Closed Sources (2)	Stage Capabilities (6)	Supply Chain Compromise (3)	Scheduled Task/Job (5)	Create Account (3)	Escape to Host	Deploy Container	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Fallback Channels	Exfiltration Over Web Service (2)	Endpoint Denial of Service (4)
Search Open Technical Databases (5)		Trusted Relationship	Serverless Execution	Create or Modify System Process (4)	Event Triggered Execution (16)	Direct Volume Access	Modify Authentication Process (7)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains (3)		Valid Accounts (4)	Shared Modules	Event Triggered Execution (16)	Exploitation for Privilege Escalation	Domain Policy Modification (2)	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (4)	Data from Information Repositories (3)	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			Software Deployment Tools	External Remote Services	Hijack Execution Flow (12)	Execution Guardrails (1)	Multi-Factor Authentication Request Generation	Domain Trust Discovery		Data from Local System	Non-Application Layer Protocol		Network Denial of Service (2)
			System Services (2)	Hijack Execution Flow (12)	Process Injection (12)	Exploitation for Defense Evasion	Network Sniffing	File and Directory Discovery		Data from Network Shared Drive	Non-Standard Port		Resource Hijacking
			User Execution (3)	Implant Internal Image	Scheduled Task/Job (5)	File and Directory Permissions Modification (2)	OS Credential Dumping (8)	Group Policy Discovery		Data from Removable Media	Protocol Tunneling		Service Stop
			Windows Management Instrumentation	Modify Authentication Process (7)	Valid Accounts (4)	Hide Artifacts (10)	Steal Application Access Token	Network Service Discovery		Data Staged (2)	Proxy (4)		System Shutdown/Reboot
				Office Application Startup (6)		Hijack Execution Flow (12)	Steal or Forge Authentication Certificates	Network Share Discovery		Email Collection (3)	Remote Access Software		
				Pre-OS Boot (5)		Impair Defenses (9)	Steal or Forge Kerberos Tickets (4)	Network Sniffing		Input Capture (4)	Traffic Signaling (2)		
				Scheduled Task/Job (5)		Indicator Removal (9)	Steal Web Session Cookie	Password Policy Discovery		Screen Capture	Web Service (3)		
				Server Software Component (5)		Indirect Command Execution	Unsecured Credentials (7)	Peripheral Device Discovery		Video Capture			
				Traffic Signaling (2)		Masquerading (7)		Permission Groups Discovery (3)					
				Valid Accounts (4)		Modify Authentication Process (7)		Process Discovery					
						Modify Cloud Compute Infrastructure (4)		Query Registry					
						Modify Registry		Remote System Discovery					
						Modify System Image (2)		Software Discovery (1)					
						Network Boundary Bridging (1)		System Information Discovery					
						Obfuscated Files or Information (9)		System Location Discovery (1)					
						Plist File Modification		System Network Configuration Discovery (1)					
						Pre-OS Boot (5)		System Network Connections Discovery					
						Process Injection (12)		System Owner/User Discovery					
						Reflective Code Loading		System Service Discovery					
						Rogue Domain Controller		System Time Discovery					
						Rootkit		Virtualization/Sandbox Evasion (3)					
						Subvert Trust Controls (6)							
						System Binary Proxy Execution (13)							
						System Script Proxy Execution (1)							
						Template Injection							
						Traffic Signaling (2)							
						Trusted Developer Utilities Proxy Execution (1)							
						Unused/Unsupported Cloud Regions							
						Use Alternate Authentication Material (4)							
						Valid Accounts (4)							
						Virtualization/Sandbox Evasion (3)							
						Weaken Encryption (2)							
						XSL Script Processing							

Last modified: 01 April 2022