RUTGERS UNIVERSITY

Department of Computer Science

Computer Security

Exam 3

April 22, 2024

Discussion

Prepared with tender loving care by Paul Krzyzanowski

100 POINTS – 25 QUESTIONS – 4 POINTS EACH – For each statement, select the *most* appropriate answer.

1. The main challenge with application sandboxing by *interposing system calls*, as Janus does, is:
   (a) Applications bypassing the filter and invoking system calls directly.
   (b) Accurately mirroring the current state of the operating system.
   (c) An inability to define per-app restrictions for specific system calls.
   (a) The sandbox does not know how other mechanisms, like capabilities, are configured.

The user-level process that receives the interposed system calls must be aware of TOCTTOU issues and know all the possible side effects of system calls and asynchronous events. For example, file names, sizes, or permissions may change spontaneously; network operations require multiple steps that must be tracked; and open file descriptors may be shared among processes (or not).

(a) This shouldn't be an issue if the interposing mechanism is well implemented. In Janus, it's a kernel module that intercepts every system call and then sends a request to the user-level validation process.

(c) The purpose of any sandbox is to define per-app restrictions on how system calls can be used.

(d) Those mechanisms should not affect the proper functioning of the sandbox. They might enable access to certain privileged system calls but that would have to be explicitly configured and the sandbox can still be configured to reject them if needed.

2. Seccomp-BPF (SECure COMPuting - Berkeley Packet Filter) adds to control groups, capabilities, and namespaces by:
   (a) Allowing certain processes to have access to privileged system calls.
   (b) Enabling specific non-privileged system calls to be disallowed on a per-process basis.
   (c) Giving a group of processes a private set of process IDs and user IDs.
   (d) Filtering all network traffic going into and out of the computer.

While capabilities deal with controlling access to privileged operations and namespaces deal with providing private or shared access to various identifiers and file system mount points, Seccomp allows rules (filters) to restrict any system calls on a per-process basis as well as restrict them based on their parameters (with the restriction that it cannot dereference pointers, so it cannot examine strings).

(a) is a feature of capabiliites; (c) is a feature of namespaces; (d) is a feature of a packet filter, but Seccomp-BPF is not a packet filter. The BPF in the name is there because they use the logic of the Berkeley Packet Filter and applied it to filtering system calls.

3. Malicious firmware in a USB flash drive is most likely to:
   (a) Copy files to the computer without the user being aware.
   (b) Reconfigure the bootloader or system UEFI (or BIOS).
   (c) Inject malware into files copied to the drive.
   (d) Cause the device to masquerade as a keyboard.

(a) Typically requires software running at the operating system level rather than malicious firmware within the USB device itself.

(b) While reconfiguring the bootloader or system UEFI may be possible with certain types of USB firmware attacks, particularly those that can exploit vulnerabilities at the firmware level, this attack is extremely uncommon (or nonexistent).

(c) Injecting malware involves manipulation at the file system level, which is not directly a function of firmware.

(d) Causing the device to masquerade as a keyboard (or another USB Human Interface Device) is the most common USB firmware attack. It leverages the ability of USB firmware to reprogram the device to act as a different type of device, like a keyboard, and send keystrokes to the computer, which can be used to execute malicious commands.

4. Which best describes the danger posed by a malicious *hypervisor*?
   (a) It allows an attacker to bypass authentication and run shell commands directly.
   (b) It can reinstall itself into the operating system whenever the system reboots.
   (c) It modifies critical libraries and commands to escape detection.
   (d) The operating system and all files and applications can remain unmodified, making it hard to detect.

(a) This describes describes malware that operates within the operating system rather than a hypervisor, which functions at a lower level.

(b) This is a persistence mechanism common to rootkits and bootkits but doesn't indicate hypervisor attacks.

(c) This is an aspect common to malware at the operating system level and is a core part of rootkits.

(d) A malicious hypervisor operates below the operating system, thus allowing it to monitor all interactions with the hardware by operating systems without modifying any part of the operating system or file system. This makes it extremely difficult to detect since the operating system and its components can remain entirely unmodified, but interactions with any system hardware will be under the control of the hypervisor.

5. One aspect of *Trojan Horses* that makes it different from other malware is that this malware:
   (a) Searches for vulnerabilities in other systems and propagates without human intervention.
   (b) Deletes files on the computer it infects.
   (c) Is usually installed willingly by the user.
   (d) Exploits vulnerabilities to get elevated privileges.

(a) This is behavior that is indicative of worms, not Trojans.

(b) A Trojan may delete files but this is not a distinguishing feature of a Trojan.

(c) This is the key characteristic of Trojan Horses. They typically masquerade as legitimate software, tricking users into installing them.

(d) Many types of malware exploit vulnerabilities to get elevated privileges. This is neither a differentiator of Trojans nor a common aspect of their behavior.

6. What key lesson can be derived from Ken Thompson's paper, *Reflections on Trusting Trust*?
   (a) The integrity of a system's compilers can be compromised, leading to a compromised software supply chain.
   (b) Software source code should be audited for it to be considered trustworthy.
   (c) Antivirus software should be used to defend against possible vulnerabilities in applications.
   (d) Regular updates and patches are necessary for maintaining system security.

Ken Thompson describes how a compiler can be modified to include malicious code in the programs it compiles, including in compiling the compiler itself, thereby perpetuating the attack invisibly across generations of software and hiding the code from the source. Hence, a compromised compiler can compromise code that it compiles, including itself.

(b) While auditing source code is a good security practice, Thompson's paper highlights a deeper issue: even if the source code is audited and found to be secure, the compiler used to turn that source into executable code can introduce vulnerabilities.

(c) This is generally good security advice but is unrelated to the concerns raised in Thompson's paper about the trust in the tools used to create software.

(d) This is also good security practice but does not address the issue of trusting the foundational tools in software development.

7. Which statement best differentiates *phishing* from spear phishing?
   (a) Phishing is a type of malware, whereas spear phishing is a social engineering technique.
   (b) Phishing is typically conducted through email, while spear phishing uses social media.
   (c) Phishing attacks are generic and aim at a wide audience, while spear phishing is highly targeted and customized.
   (d) Spear phishing attacks are less sophisticated and easier to identify than phishing attacks.

Phishing attacks usually target large groups of people with the hope that some will fall for the bait. In contrast, spear phishing involves tailored attacks directed at specific individuals or organizations, making them more personal.

(a) This statement is incorrect. Both are types of social engineering attacks and not malware.

(b) Either attack can use email or social media.

(d) Spear phishing attacks are usually more sophisticated because they are targeted and involve more preparation.

8. What is the primary purpose of a malware *packer*?
   (a) To encrypt the payload to avoid detection by security software.
   (b) To distribute malware to unsuspecting users.
   (c) To integrate multiple types of malware into a single payload for easier distribution.
   (d) To give attackers an interface for configuring malware functions before deploying it.

A malware packer is software that is used primarily to obfuscate or conceal malware from detection mechanisms such as antivirus software by transforming the malicious code. By encrypting or otherwise obfuscating the code, the packer makes it difficult for security software to analyze and detect the underlying malicious content.

9. A *bootkit* differs from a rootkit because a *bootkit*:
   (a) Is a tool used for the secure loading of operating systems, while a rootkit is malware.
   (b) Exploits vulnerabilities in a program and obtains elevated privileges.
   (c) Hides in the operating system and gives an attacker access to the system.
   (d) Gets control before the operating system loads.

(a) is incorrect. (b) describes the characteristic of rootkits. (c) is true for rootkits and doesn't differentiate between rootkits and bootkits.

(d) Describes the key distinguishing feature of bootkits. Bootkits are designed to load and execute before the operating system starts, allowing them to take control of the system at a very early stage. This enables them to bypass many security mechanisms that would normally be active once the operating system loads.

10. What is the purpose of a *bot* in a botnet?
   (a) To serve ads to track user activity.
   (b) To perform tasks assigned by a command-and-control server.
   (c) To monitor incoming traffic to protect systems from malware.
   (d) To upload virus signatures to a server whenever malware is detected on a system.

(a) Serving ads or tracking user activity may be tasks assigned to a bot but are not common purposes.

(b) This is the core function of a botnet. Typical tasks include launching denial of service (DoS) attacks, sending spam email, executing fraudulent transactions, or mining cryptocurrency.

(c) This is incorrect. Bots are used for malicious purposes. (d) This is also incorrect.

11. A CAM *overflow* attack works by:
    (a) Forcing a switch to direct all traffic to a specific port on the switch.
    (b) Using an unauthorized port to fill up the switch table.
    (c) Getting a switch to forget the stored association between MAC addresses and switch ports.
    (d) Overflowing the switch table, causing the switch firmware to crash and reset.

(a) This describes a potential consequence of another type of attack, like ARP spoofing, rather than a CAM overflow attack.

(b) While this describes adding entries to a switch table, the phrase "using an unauthorized port" is not specifically relevant to how CAM overflow attacks work.

(c) This describes the purpose of causing the overflow: to cause the switch to forget which port corresponds to which MAC address. This makes the switch behave like a hub, where it will send received packets out to all ports.

(d) The attack indeed overflows the switch table (obvious from the name of the attack) but an overflow does not result in the switch crashing.

12. Which of the following is a common security issue associated with both ARP (Address Resolution Protocol) and DHCP (Dynamic Host Configuration Protocol)?
    (a) They rely on mutual authentication between the client and server.
    (b) A malicious client can contact either service without authentication or authorization.
    (c) A DNS rebinding attack can cause clients to identify themselves incorrectly.
    (d) A client has no way of knowing who the authoritative server is for either service.

(a) is incorrect: neither ARP nor DHCP rely on mutual authentication.

(b) While a malicious host can contact these services, the problem isn't about contacting them but rather how these protocols can be manipulated.

(c) DNS rebinding attacks are not related to ARP or DHCP protocols.

(d) This is a vulnerability common to both ARP and DHCP. ARP does not have a mechanism to verify the identity of the host responding to ARP requests, which makes it susceptible to ARP spoofing/poisoning. Similarly, DHCP clients have no built-in method to tell whether the DHCP server responding to their requests is legitimate, which can lead to DHCP spoofing.

13. What is the main purpose of using *SYN cookies* in TCP communication?
    (a) To validate user credentials.
    (b) To speed up the connection establishment.
    (c) To not have to store information from the first connection message.
    (d) To send information about existing authenticated sessions.

SYN cookies are a technique used to prevent SYN flood attacks, a type of Denial of Service (DoS) attack.

(a) There are no user credentials to validate.

(b) They do not speed up connection setup. More likely, they will add a small overhead due to the computation needed to generate and then verify the cookies (i.e, compute a hash).

(c) Normally, a server must allocate memory when it receives a SYN request. SYN cookies allow the server to avoid storing state information by encoding the state of the connection and a server secret into a hash that will be sent as an initial sequence number (ISN) in the SYN-ACK response. This way, if the client responds with an ACK, the server can reconstruct and validate the sequence number and compare it with the cookie provided in the ACK, significantly reducing the impact of SYN flood attacks, which would have unroutable or random source addresses.

(d) SYN cookies are not related to session information or any authentication state of sessions.

14. Which malware mechanism involves *altering the DNS settings* on a victim's device?
    (a) Email spoofing.
    (b) Adware.
    (c) Credential stuffing.
    (d) Pharming.

(a-c) do not involve changing DNS settings.

Pharming involves redirecting a user's traffic to fraudulent servers by manipulating DNS settings, either on the user's device or at the DNS server. This can be achieved by installing malicious software on the user's device that changes the local DNS settings, modifying the user's local hosts file, or by attacking DNS servers directly to corrupt their address resolution functions.

15. Which of the following best describes a common attack on the *Border Gateway Protocol* (BGP)?
    (a) False DNS information is injected into the network.
    (b) Internet traffic is redirected by advertising false route information.
    (c) The network is flooded with excessive traffic to overwhelm it.
    (d) Buffer overflow attacks that cause BGP routers to crash.

(a) DNS injection is an attack that targets DNS systems and does not involve BGP.

(b) This attack is called *BGP hijacking* or *route hijacking.* In this attack, a malicious actor sends false routing information to BGP routers, which can redirect internet traffic through an attacker-controlled router, allowing for interception or disruption of the data flow.

(c) This is a DDoS attack.

(d) This is neither a common nor specifically targeted attack on BGP.

16. How does a *DNS rebinding* exploit affect the same-origin policy?
    (a) It strengthens the same-origin policy by frequently updating DNS records.
    (b) It bypasses the same-origin policy by allowing scripts to access data from different domains without restriction.
    (c) It circumvents the same-origin policy by changing the DNS resolution after the initial page load.
    (d) It disables the same-origin policy entirely, allowing unrestricted cross-site scripting.

(a) This is wrong. DNS rebinding does not strengthen the same-origin policy; rather, it exploits it.

(b) While DNS rebinding can lead to cross-domain requests, it doesn't inherently allow scripts to access data from different domains without any restrictions.

(d) DNS rebinding does not disable the same-origin policy entirely. Instead, it cleverly manipulates it to achieve similar outcomes under specific conditions.

(c) DNS rebinding involves changing the DNS resolution for a domain to an attacker-controlled IP address (for example, an attacker address or an address within the victim's private network) after the initial page has loaded. As a result, any subsequent requests to the domain can be directed to a new location, even though, to the browser, they appear to originate from the same trusted domain, Because of this, it circumvents the same-origin policy.

17. Transport Layer Security (TLS) differs from Virtual Private Networks (VPNs) because TLS:
    (a) Provides end-to-end encryption only between communicating applications.
    (b) Secures all data flowing between two networks.
    (c) Secures all data flowing between two machines.
    (d) Is designed exclusively for the web and secures HTTP communications.

(a) TLS encrypts the connection between two applications, ensuring that the data exchanged remains confidential and adds a MAC to ensure data is unaltered during transmission. Because it operates at the transport layer, this encryption and integrity checking applies to the data being exchanged by the applications (such as a web browser and a web server), rather than the entire network.

(b) This describes a function of a VPN more accurately than TLS. VPNs create a secure and encrypted tunnel between two networks, securing *all* traffic that passes through this tunnel, regardless of the type of application generating the traffic.

(c) While TLS does secure communications between machines, it's more specific to the applications on those machines rather than securing all data between the machines. VPNs are better suited for the role described in this option.

(d) Although TLS is widely used to secure HTTP communications (via HTTPS), its use is not exclusive to the web. TLS can also secure other types of communications, such as messaging apps and email. Common email sending & access protocols, like STP, POP, and IMAP, all usually run over TLS (SMTPS, POPS, and IMAPS).

18. An assumption in a *zero-trust architecture* is:
    (a) Computers within a local area network can be trusted, but those outside cannot.
    (b) We cannot rely on firewalls to block malicious network traffic.
    (c) Computers offering internet-facing services should be placed in a different subnet than other systems.
    (d) Remote users should use a VPN to connect to a corporate network.

The principle of a zero-trust architecture is that organizations should not automatically trust something just because it is within a perimeter (e.g., within a LAN protected by a firewall). Mobile devices leave perimeters and return; users unwillingly download malware; malware searches out other systems. With zero-trust, every connection should be authenticated and authorized.

(a) This contradicts the fundamental premise of zero-trust architecture,

(b) While this does not capture the full scope of zero trust with its need for verification, this points out a key driver behind the architecture: that there is no implicit trust and we cannot count on perimeter defenses to filter out the bad stuff.

(c) This points to a practice of network segmentation, where groups of machines are in isolated networks. In this case, the segmentation is setting up a DMZ (demilitarized zone). Segmentation is a component of enhancing security but does not directly speak to the zero-trust principle of not inherently trusting any entity just because it's within a perimeter.

(d) The use of a VPN is common for securing remote connections. In the case of a host-to-network VPN, it essentially brings a remote system into the internal, trusted network. Zero-trust architecture goes beyond secure connections and encompasses a more holistic approach where trust is never, regardless of the network connection method.

19. Which technology is best suited for *deep content inspection*, such as examining the contents of an email attachment?
    (a) Screening router.
    (b) Stateful packet inspection firewall.
    (c) Application proxy.
    (d) Transport Layer Security.

(a) A screening router is typically used to create a basic barrier and packet filter between networks. It can enforce rules based on IP addresses and ports. However, it does not have the capability to inspect content deeply, especially within application data like email attachments.

(b) Stateful packet inspection (SPI) firewalls monitor the state of active connections and can block packets that don't match known, safe configurations. They're a step above a simple packet filter but SPI firewalls do not typically look at application layer content.

(c) An application proxy is an intermediate device for requests from clients seeking resources from other servers. It is capable of understanding application-level protocols (such as HTTP, FTP, SMTP) and can inspect and filter content at a high level of detail. This includes examining the contents of files and attachments in emails, making it suitable for deep content inspection.

(d) While TLS secures data transmission, it does not inspect or filter content. Its role is encryption and secure transmission, not content analysis.

20. A key feature of a Distributed Denial of Service (DDoS) *reflection attack* is:
    (a) Sending messages that result in large responses from the victim.
    (b) Sending requests with a single spoofed source address to a third-party service.
    (c) Overwhelming a system with high volumes of data from multiple sources.
    (d) Exploiting malware to cause a network service to crash on the victim's machine.

(a) This describes a general amplification attack but doesn't directly address the reflection component, which involves a third party.

(b) This describes the essence of a DDoS reflection attack. In this type of attack, the attacker sends a large number of UDP requests to a third-party server (such as a DNS or NTP server) with a spoofed source IP address that is actually the victim's IP address. This causes the third-party server to send responses to the victim, not knowing that the initial requests were fraudulent and came from somewhere else.

(c) This describes a distributed denial of service (DDoS) attack but does not focus on reflection.

(d) This is unrelated to the mechanisms of a reflection attack.

21. Which URL shares the same origin as `https://www.cs.rutgers.edu/index.html`?
    (a) `https://www.cs.rutgers.edu/error-page`
    (b) `http://www.cs.rutgers.edu/index.html`
    (c) `https://cs.rutgers.edu/index.html`
    (d) `https://www.cs.rutgers.edu:8080/index.html`

An origin is defined by the scheme (protocol), hostname (domain), and port number.

(a) Shares the same scheme (https), hostname (www.cs.rutgers.edu), and port number (443, implicit because the scheme is https).

(b) Different scheme (http vs. https) and different port (the default port for http is 80).

(c) Different hostname: www.cs.rutgers..edu is different from cs.rutgers.edu.

(d) Different port: 8080 is explicitly specified here.

22. The web's *same-origin policy*:
    (a) Does not allow a website to load content from other sites.
    (b) Prevents a web page from running scripts loaded from other websites.
    (c) Prevents scripts on a web page from reading or writing content loaded from other websites.
    (d) Treats all frames within a web page as having the same origin as the main URL of the page.

(a) This is incorrect. Websites frequently load content from other sites, such as images, scripts, and stylesheets. The same-origin policy does not prevent these actions outright; it controls access to potentially sensitive data across origins.

(b) This is incorrect. Web pages can run scripts loaded from other websites. For example, many websites include third-party JavaScript libraries or analytics scripts from external sources.

(c) This is correct. The same-origin policy prevents scripts from accessing or modifying data (such as the DOM of a document) that originates from a different domain, scheme, or port, unless explicitly allowed (e.g., via CORS: Cross-Origin Resource Sharing).

(d) This is incorrect. Each frame in a web page is treated according to its origin. A frame from a different origin than the main page will be restricted by the same-origin policy.

23. Which of the following best describes a cross-site scripting (XSS) attack?
    (a) An attacker exploits vulnerabilities in web applications to inject malicious scripts into pages viewed by other users.
    (b) An attacker directly infects a website's database with malicious SQL queries.
    (c) An attacker uses email phishing to steal users' login credentials.
    (d) An attacker floods a web server with excessive requests to make it unavailable to users.

XSS attacks enable attackers to inject malicious scripts into content that other users see.

(a) XSS involves exploiting vulnerabilities in web applications, allowing the attacker to inject client-side scripts into web pages viewed by other users. These scripts can then execute within the context of the victim's session, potentially leading to unauthorized actions, data theft, or other malicious outcomes.

(b) This describes an SQL injection attack, not XSS.

(c) This describes phishing, a form of social engineering.

(d) This describes a Distributed Denial of Service (DDoS) attack.

24. What is *clickjacking* in the context of web security?
    (a) A technique where an attacker uses rapid clicks to overload a web server, similar to a DDoS attack.
    (b) A method where users are redirected to malicious websites through hyperlinks that appear legitimate.
    (c) JavaScript that generates lots of fake clicks from a browser for social media likes or increasing website traffic.
    (d) An attack where the victim is tricked into clicking on something different from what the user perceives?.

Clickjacking is a malicious technique used by an attacker to trick a user into clicking on something different from what the user perceives, typically by overlaying a transparent layer over what appears to be legitimate web page content.

25. A defense against *reflected cross-site scripting* is:
    (a) Sanitizing inputs.
    (b) Enforcing the same-origin policy.
    (c) Disabling Cross-Origin Resource Sharing (CORS).
    (d) Using TLS to create encrypted sessions.

Reflected cross-site scripting (XSS) occurs when an application receives data in an HTTP request and includes that data in the HTTP response in an unsafe way. For example, if user input from a URL query is directly included in web page content without adequate sanitization, it can be exploited to execute malicious scripts. Sanitizing inputs is one of the most effective defenses against any form of XSS, including reflected XSS. Sanitizing inputs involves cleaning and filtering all incoming data to ensure that it does not contain executable code before it is rendered back to the user.