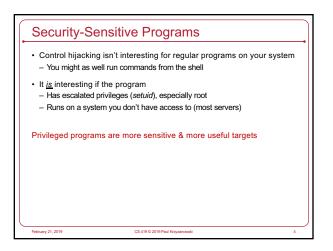
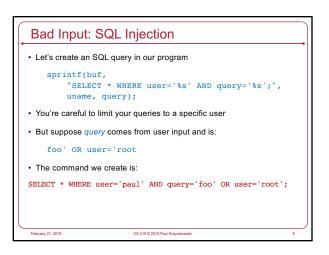


Last week, we looked at ... Defenses • Programming languages with bounds checks & strong typing - Use "safe" functions in C/C++ - Java, C# - Python is vulnerable in some areas • But native methods might be vulnerable • Data execution protection (DEP) no-execute memory pages for stack & heap - Attacks: return-to-libc or Return-Oriented-Programming attacks • Address Space Layout Randomization (ASLR) - Attacks: • not all programs or libraries use ASLR • NOP slod - create a huge block of NOPs to increase chance of jumping to exploit • Try and try again if there isn't much entropy in the randomization • Stack canaries • Attack: if canary is modified, the compiler causes an exception. If you can modify the exception handler, it can point to your code: Structured Exception Handling (SEH) exploit.

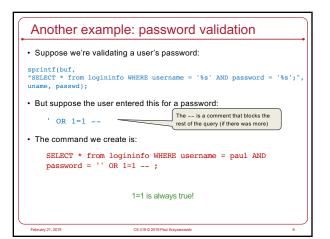


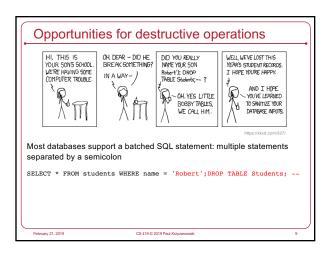
Injection attacks Injection is rated as the #1 software vulnerability in 2017 by the Open Web Application Security Project (OWASP) Index an attacker to inject code into a program or query to Execute commands Modify a database Change data on a website We looked at buffer overflows and format strings ... but there are other forms too

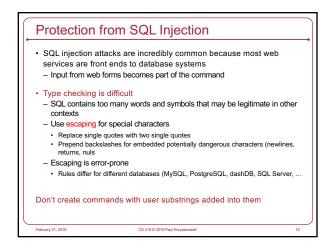


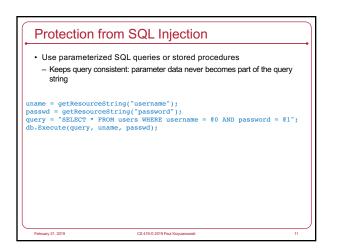
What's wrong?

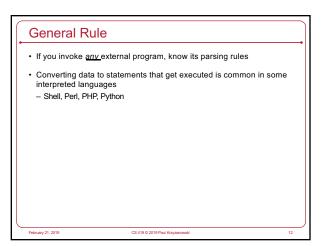
- We should have used *snprintf* to avoid buffer overflow (but that's not the problem here)
- We didn't validate our input - And ended up creating a query that we did not intend to create!

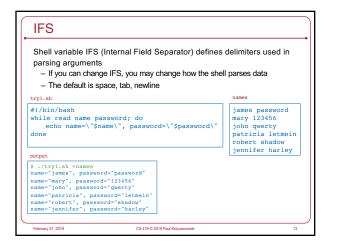


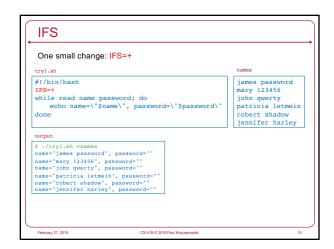


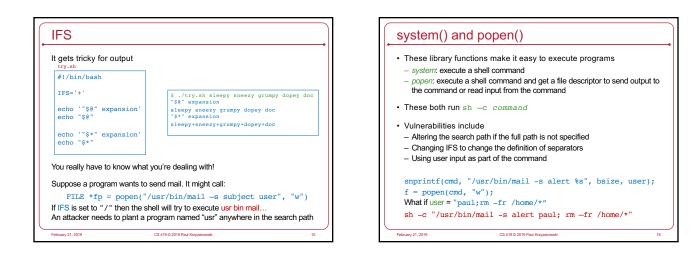


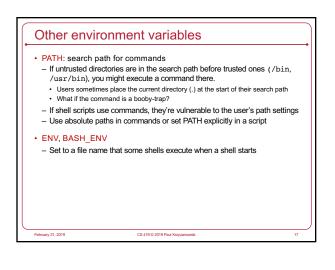


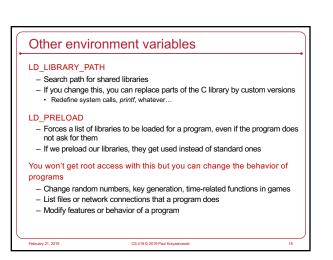


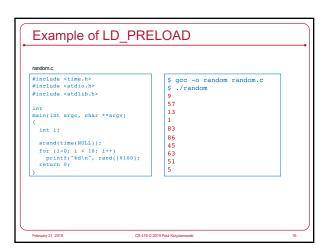


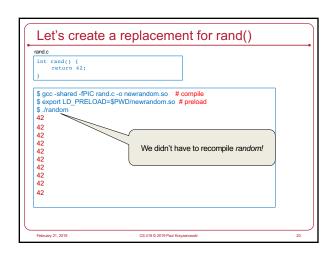


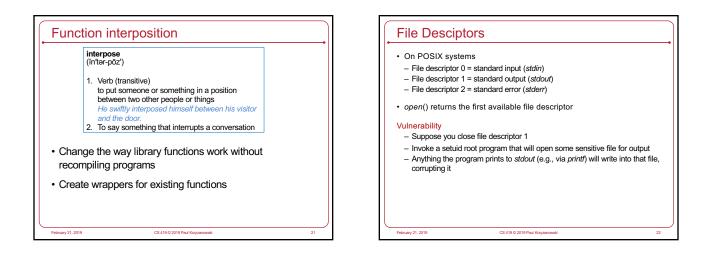


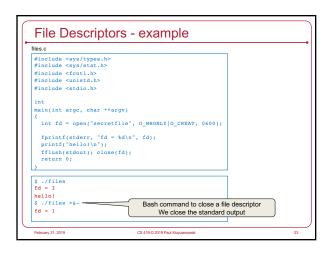




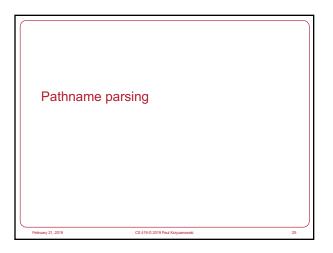


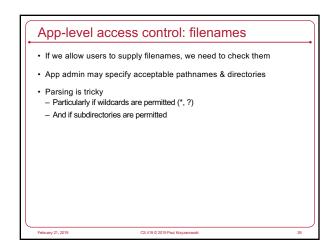






/indows CreateP	rocess function	
_In_opt_ _In_ _In_ _In_opt_ _In_opt_ _In_	LPCTSTR LPTSTR LPSECURITY_ATTRIBUTES BOOL DWORD LPYOID LPCTSTR LPSTARTUPINFO	
10 parameters t	hat define window creation	on, security attributes, file
inheritance, and	others	





Parsing directories

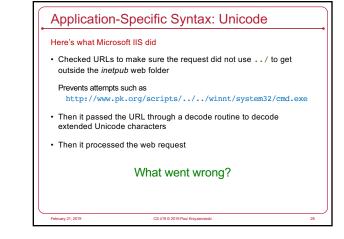
- Suppose you want to restrict access outside a specified directory
 Example, ensure a web server stays within /home/httpd/html
- · Attackers might want to get other files
- They'll put .. in the pathnaame
- .. is a link to the parent directory
- For example:

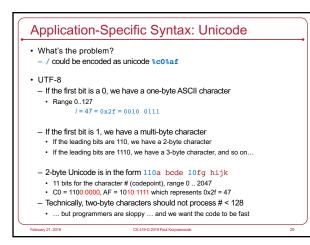
ary 21, 2019

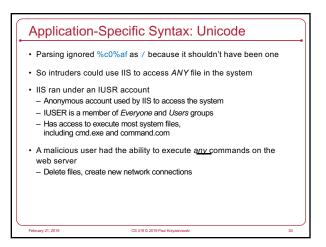
- http://pk.org/../../etc/passwd
- The ... does not have to be at the start of the name could be anywhere http://pk.org/419/notes/../../416/../../etc/passwd
- But you can't just search for . . because an embedded . . is valid $\tt http://pk.org/419/notes/some.junk..goes..here/$
- Also, extra slashes are fine
- http://pk.org/419////notes///some..junk..goes..here///

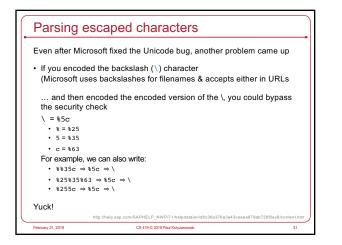
CS 419 © 2019 Paul Kr.

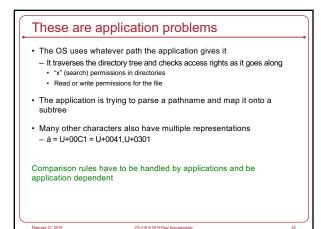
Basically, it's easy to make mistakes!

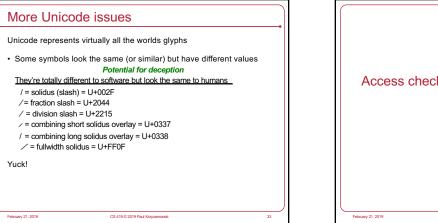


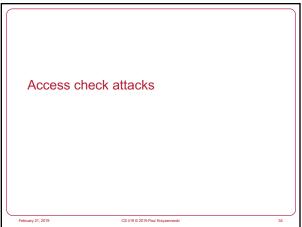


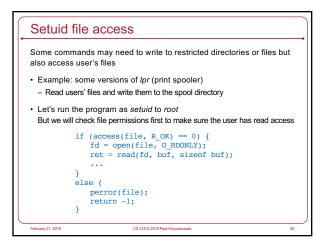


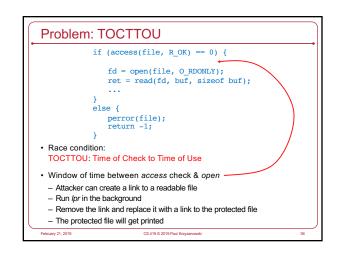


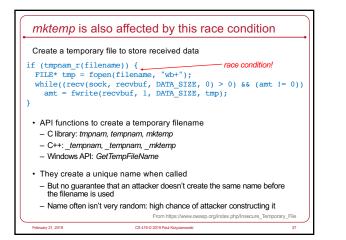


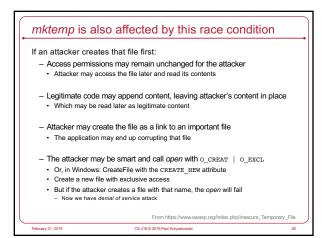












Defense against mktemp attacks

Use mkstemp

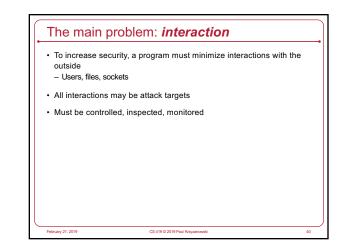
ary 21, 2019

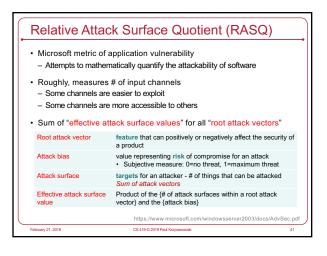
- · It will attempt to create & open a unique file
- You supply a template
 A name of your choosing with xxxxxx that will be replaced to make the
 name unique
 mkstemp("/tmp/secretfileXXXXX")

CS 419 © 2019 Paul Krz

- File is opened with mode 0600: r-- ---
- If unable to create a file, it will fail and return -1

 You should test for failure and be prepared to work around it.





RASQ Sample root vectors & bias values				
Root vector	Bias value	Comment		
Open sockets	1.0	Every open & listening socket is a potential target		
Open RPC endpoints	0.9	Like sockets but require more skill		
Enabled accounts	0.7	Default accounts simplify brute-force password attacks		
Enabled accounts in the Administrator group	0.9	Admin accounts are higher risk		
Weak ACLs in file system	0.2	Most files in the system are targeted after a system is compromised		
Weak ACLs on file shares	0.9	Default shares are commonly known and often targeted		
https://www.microsoft.com/windowsserver2003/docs/AdvSec.pdf				
February 21, 2019 CS 419 © 2019 Paul Krzyzanowski 42				

Summary • Better OSes, libraries, and strict access controls would help

- A secure OS & secure system libraries will make it easier to write security-
- sensitive programs

 Enforce principle of least privilege
- Validate all user inputs ... and try to avoid using user input in commands

Reduce chances of errors

ary 21, 2019

- Eliminate unnecessary interactions (files, users, network, devices)
- Use per-process or per-user /tmp
- Avoid error-prone system calls and libraries
- Or study the detailed behavior and past exploits
 Minimize comprehension mistakes
- Specify the operating environment & all inputs
- And validate or set them at runtime: PATH, LD_LIBRARY_PATH, user input, ...
 Don't make user input a part of executed commands

CS 419 © 2019 Paul Krzyzz

