VULNERABILITY DECODER

Race Condition

The Vulnerability

A race condition vulnerability occurs when a system that's designed to handle tasks in a specific sequence is forced to perform two or more operations at the same time. Without proper controls, different processes may interfere with each other and create an opening for an attack.

The Risks

Race condition attacks are relatively difficult to engineer, but once an intruder breaches a system, it's possible to alter, manipulate, or steal data, change privileges, insert malicious code, unleash a denial of service attack, and deactivate security controls.

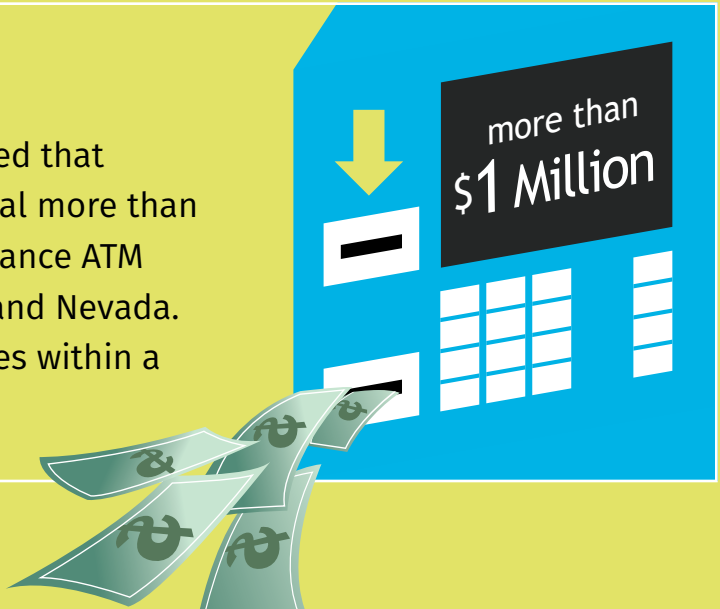
8.5%

of applications have a race condition vulnerability on initial scan.

Source: SOSS v11

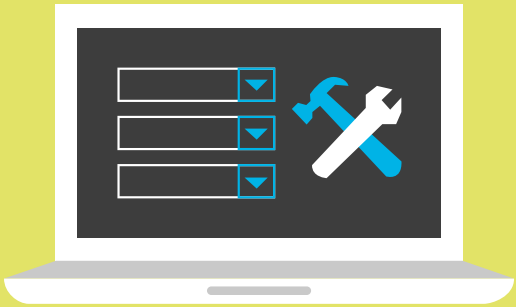
Example Breach

In one high-profile case, the FBI reported that attackers used this methodology to steal more than $1 million from Citibank using cash advance ATM kiosks at casinos located in California and Nevada. The attackers sent near identical queries within a 60-second time window.



Prevention & Remediation

Race condition attacks are preventable with secure coding practices. It's critical to scan and review code for race condition vulnerabilities. This includes the use of static analysis.



This is an example of a race condition vulnerability:

```
/* vulp.c */
#include
#include
#define DELAY 10000
int main()
{
char * fn = "/tmp/XYZ";
char buffer[60];
FILE *fp;
long int i;
/* get user input */
scanf("%50s", buffer );
if(!access(fn, W_OK)){
/* simulating delay */
Laboratory for Computer Security Education 2
For (i=0; i < DELAY; i++){
int a = i^2
}
fp = fopen(fn, "a+");
fwrite("\n", sizeof(char), 1, fp);
fwrite(buffer, sizeof(char), strlen(buffer), fp);
fclose(fp);
}
else printf("No permission \n");
```

Recommendations

Nobody writes perfect code the first time around. You can avoid vulnerabilities and prevent breaches when you:

- ☑ Get training in secure coding best practices through on-demand eLearning courses, in-person security consultations, and professional development certifications and conferences.
- ☑ Scan early and often to detect flaws while you code. Use application security tools that allow you to scan small batches of code instantaneously, and provide remediation guidance within your development workflow.

Download the Secure Coding Best Practices Handbook

Join the Veracode Community
community.veracode.com

VERACODE