CERT'S PODCASTS: SECURITY FOR BUSINESS LEADERS: SHOW NOTES

Managing Security Vulnerabilities Based on What Matters Most

Key Message: Determining which security vulnerabilities to address should be based on the importance of the information asset.

Executive Summary

Attacker objectives and the means they use to exploit software vulnerabilities are constantly changing. CERT typically sees anywhere from 7000 to 8000 new vulnerability reports every year. Even if it was possible, it does not make good business sense to address every software vulnerability. Business leaders need to better understand which vulnerabilities are worth their attention based on which information assets are most critical to the business. These assets are the ones that require the greatest level of protection as soon as a new vulnerability is disclosed.

In this podcast, Art Manion, the leader of CERT's Vulnerability Analysis Team, discusses how security vulnerabilities have evolved and how business leaders can more effectively manage the growing number of new ones.

PART 1: THE CHALLENGES IN DEFINING A SECURITY VULNERABILITY

Some Definitions

A software vulnerability is most often some type of coding defect or bug that allows an attacker to break into a system or violate a security policy.

Elements of this definition include:

- conditions that allow for potential impact or attacker gain
- a consequence that is not typically permitted
- a violation of an explicit security policy, for example, a firewall rule that states "this port or service is not allowed to cross this boundary."
- something caused by a human – an engineer, a designer, a developer
- failure to configure the software properly or configure it to be vulnerable
- placing secure software in a vulnerable, changing environment, for example, having such software use protocols that have no notion of authentication or encryption

An example of the last bullet occurs in SCADA (supervisory control and data acquisition) systems that use protocols such as DNP3 or Modbus. These protocols were never designed with security in mind.

PART 2: THE SHIFTING VULNERABILITY LANDSCAPE

Vulnerability Trends

Many of today's vulnerabilities are due to the prevalence of web-based software – web applications, software-as-a-service, and Web 2.0 to name a few.

Attacks include cross-site scripting, SQL injection, and cross-site request forgery.

Historically, operating system services were the typical targets of attack – including UNIX and Windows-based

network services.

Today, attackers are much more interested in targeting personal information – credit card numbers, bank accounts, passwords, and access to websites. Phishing is a common approach.

Web browsers, browser plugins, and media players are increasingly targets of attack. Attackers solicit users to visit a website, click on a spam or phishing link, or open an email attachment, and then use this action to install keyloggers, perform screen capture of userids and passwords, install malicious code, and upload data of interest for later use.

Consequences – Targeted vs. Widespread Attacks

The key point to remember is that when an attacker has gained access to any computer – home or business – the attacker can do anything that an authorized user can do. For businesses, this includes the potential to access sensitive databases and applications as well as being able to send email that appears legitimate.

Attackers can target a specific organization or set of computers for a specific purpose or cast their net broadly, seeking to add compromised computers to their growing botnet. (Refer to Tackling the Growing Botnet Threat podcast for more information about botnets.)

PART 3: DETERMINING WHICH VULNERABILITIES TO PAY ATTENTION TO

Finding Out About Current Vulnerabilities

Most new vulnerabilities – 7000-8000 per year – are publicly reported on mailing lists, blogs, and websites.

There is a fairly mature community that watches for vulnerability reports, catalogs them, and provides brief descriptions – such as US-CERT.

Responsible Disclosure

Often vulnerabilities are reported directly to the vendor by researchers. This is called "responsible disclosure" as it gives the software producer a chance to fix the problem before it is publicly disclosed.

Vulnerabilities can be easily discovered by using automated tools such as fuzz testers and source code analysis tools.

Zero-Day Vulnerabilities

A zero-day vulnerability is only known by a small community, such as the researcher who discovered it. Based on this, it can sometimes have monetary value to a potential buyer. There is a market where such information is traded. Once the vulnerability becomes more well known, its value decreases.

Where to Pay Attention

Automated support is essential for paring down the number of vulnerabilities that warrant attention. This support needs to reflect which computing and information assets are most important to the business, thus the ones most worthy of protecting.

For example, if there is a key database that is used by business-critical applications, this database is a high value asset. Any new database software vulnerabilities should be assigned a higher severity level – which makes them higher priority to address quickly.

If another business unit does not use this database, they shouldn't waste time on analyzing vulnerability reports for it.

Decision Support

Ideally, an organization needs some type of expert-based, decision support system that rates vulnerabilities according to priority level – for example identifying the 3 or 4 highest priority vulnerabilities out of the latest 20, for further analysis. Analysts are immediately assigned to these. They address the remaining vulnerabilities as time permits.

Developing a vulnerability information management system is part of CERT's current research efforts.

Getting in Front of Software Vulnerabilities

While alerting and filtering services are helpful, these are more reactive solutions and are never ending.

What's needed is software that doesn't have as many vulnerabilities in the first place. Many studies have shown that it is much more cost effective to fix a defect early in the software development life cycle rather than in production.

CERT's Secure Coding Initiative, as one example, is working on developing a secure coding standard for C and C++. Code analysis tools are increasingly being used to help identify vulnerabilities during development.

Future Direction

Tying a discovered vulnerability to a secure coding rule and then moving such a rule into an accepted standard are steps toward sustainable improvement – and a defensible return-on-investment.

Resources

CERT's Vulnerability Remediation web site and blog

CERT's Secure Coding Initiative

Burch, Hal; Manion, Art; Ito, Yurie. "VRDA: Vulnerability Response Decision Assistance." CERT, JPCERT CC, Presentation at 19th Annual FIRST Conference, Seville, Spain. June 17-22, 2007.

Burch, Hal; Manion, Art; Ito, Yurie. "Vulnerability Response Decision Assistance." European Conference on Computer Network Defense (EC2ND), Heraklion, Greece. October 4-5, 2007.