HUNTING FOR WEAK CRYPTO
VULNERABILITY AND RISK MANAGEMENT

ABSTRACT

Poor implementations of cryptography and weak algorithms threaten businesses by lulling them into a false sense of security. Attacks against cryptography are particularly insidious because data can be decrypted passively and offline. Lack of enterprise awareness makes the case for a tool that inventories all software on a network that uses cryptography and highlights systems and components that either poorly implement crypto or run at weak and outdated levels of security.

Today's security best practices are tomorrow's security vulnerabilities. Cryptography is key to secure system design but it ages over time as researchers analyze algorithms and reveal weaknesses in the underlying math that can be exploited to obtain private and sensitive data. Software and devices that use outdated cryptography are prevalent within many organizations, some of these systems were constructed with cryptographic vulnerabilities and some systems age and develop cryptographic vulnerabilities over time. Often these security issues occur because:

Cryptography Challenge #1
ALGORITHM OBSOLESCENCE

New systems are rarely designed with state-of- the-art cryptography due to compatibility requirements. Over time new attacks against previously chosen algorithms and key sizes increase risk.

Cryptography Challenge #2
SUPPLY CHAIN RISK

Frequently there are multiple layers of supply chain relationships between the cryptography implementers, and the end user whose data needs to be protected. Cryptographic weaknesses in the system components can be hidden from the enterprise.

Cryptography Challenge #3
IMPLEMENTATION ERROR

Software developers misconfigure cryptographic algorithms. Often they make poor selections relating to modes, key sizes, and the need for things like countermeasures. (e.g. timing attacks, or sources of entropy)

Cryptographic algorithms are complex; often these algorithms have a number of parameters that are set and standardized to ensure compatibility across devices and applications and give a choice with regards to security levels that are offset by speed requirements. However, software programmers sometimes write code in a way that inadvertently weakens the security level of the chosen cryptographic algorithm while still appearing to be adequate in strength. For instance, seeding a random number generator with the system date when performing a key agreement; it appears to encrypt and decrypt adequately, but in fact this mistake allows an adversary to easily guess secret keys. If these implementation flaws are discovered later during manual security reviews, and the vendor is cooperative, then they are quickly corrected, and a patch is issued to fix the problem. However, if a system administrator does not install the patch then these implementation flaws can exist and propagate within a software application or network for a very long time.

There are many free cryptographic libraries available on the internet of varying quality. Sometimes software developers will indiscriminately copy a cryptographic implementation from an Open Source project and include it in their software application. Then later, the Open Source project is discovered to have a weak algorithm implementation and so it is corrected and updated. However, that fix is often missed or ignored and the vulnerable software application is never upgraded.

All of these issues make it difficult for organizations to track cryptography that is being employed to protect sensitive data and communication between applications.

CHALLENGE OF DISCOVERING CRYPTOGRAPHY IN SOFTWARE

It is very difficult to determine if a piece of software is using the correct cryptographic implementations because the software is compiled and the source code is often not available to review. In addition, there are a number of places where the cryptography could actually reside. Some software contains cryptographic algorithms built directly into the main application executable files. Other software may only link to cryptographic implementations that are part of the host operating system or the CPU. In addition, some popular security designs may require cryptographic operations to be performed in a completely separate device like a smartcard or a hardware security module. Tracking down these linkages can be very time consuming and difficult across a plethora of software and systems.

Some organizations try to manage and control all aspects of security software that run on their network using purpose built tools like a PKI. These organizations spend a lot of time performing manual security assessments on software and security features to ensure adherence to the organization's security policy. But sometimes special cases arise, and otherwise security conscious and well intentioned people deploy software applications that perform cryptography and key management that sidestep the standard organizational policy and security management infrastructure. In these cases, organizations still need to find and track this special case software, which is difficult to do because the software operates outside of the control of the company's standard security infrastructure.

SCANNING FOR CRYPTOGRAPHY

AgileScan is a security management tool that automatically generates an inventory of all cryptographic algorithms found in any piece of software on a Windows or Linux host. AgileScan is similar to an anti-virus product, but instead of searching for signatures of viruses, AgileScan searches for cryptographic indicators contained within a software binary. AgileScan then classifies the discovered algorithms to produce an enterprise cryptography inventory as well as uncover instances of old, antiquated or broken cryptography, and identifying software with publicly known cryptographic issues.

For instance, SHA-1 was a popular hashing algorithm that has reached its end of life and is no longer considered a good security choice. Old hashing algorithms like SHA-1 or MD5 should no longer be used on business critical systems because they are considered to be broken by security professionals. AgileScan will hunt for any instance of SHA-1 or MD5 found on the corporate network and report a security policy violation to staff for either remediation or acceptance.

SCANNING FOR MISCONFIGURED CRYPTOGRAPHY

Another common problem with cryptography in software is that software developers can configure or use the algorithms improperly causing a seemingly secure software implementation to be vulnerable to attacks.

For instance, secure software often requires secret keys that need to be stored on a disk. In a well-designed security system, these keys will be encrypted and stored on disk using a password, trusted platform module, or by employing an obfuscation technique. However, in a poorly designed security system it is not uncommon to find keys simply stored, unsecured on a disk drive.

Another common problem involves self-signed certificates. Certificates are used as a secure way to authenticate and communicate cryptographic keys, and they use a digital signature issued by a company's certificate authority (CA) or public-key infrastructure (PKI). Since obtaining a proper certificate can be an onerous task an often rushed administrator may simply forego proper security channels and use a certificate that is signed by themselves. Most organizations would view this as a security violation because it bypasses their security management tools. AgileScan is able to scan for self-signed certificates and report any instances back to security personnel.

HOW AGILESCAN WORKS

AgileScan has a number of scanning phases and features that work on a host file system to inventory cryptography and uncover cryptographic related problems.

KNOWN CRYPTOGRAPHIC LIBRARY DETECTION

AgileScan uses a database of known software that contains cryptographic implementations and determines if the software version is up-to-date and safe to use or outdated and containing known flaws.

DEEP BINARY STREAM INSPECTION

Cryptographic algorithms must follow precise specifications in order to operate correctly. As a result, most cryptographic algorithms can be easily identified in binary files with a high degree of certainty. This process is similar to manual inspection by an expert in forensic binary analysis, but is performed automatically by software.

CRYPTOGRAPHIC INDICATOR DATABASE

The AgileScan research and development team are constantly updating the AgileScan cryptographic indicator database and providing updates for AgileScan agents to detect additional ciphers and implementations. AgileScan uses cryptographers and crypto-implementers to continually refine this database for better accuracy and methods of broader detection.

THE BUSINESS CONTEXT OF CRYPTOGRAPHY

AgileScan reports on the business relevance of its findings. Are the identified programs part of the operating system, or a standalone binary? Is it currently running? Is it talking on the network?

COMMON CONFIGURATION FAULT DETECTION

AgileScan includes a number of specific tests for common misconfiguration faults related to cryptographic algorithm setup and usage. For example, identifying self-signed certificates which do not belong to a recognized Certificate Authority. These tests are constantly being developed and updated by the AgileScan research and development team. AgileScan searches for software that contains cryptography, identifies the cryptography being used in the software, and inventories and reports on cryptographic subroutines that were found. AgileScan is used to "hunt" for cryptography that has become antiquated and ineffective over time but which may still be relied upon in business operations due to the longevity of long-lived software often found in well-established enterprises. AgileScan gives security operations teams an easy and effective way to inventory all of the cryptographic controls that reside on hosts within their network and discover controls that are either outdated, broken or do not conform to organizational security policy.