

Cryptovirology: Threats and Countermeasures

Abstarct

We describe a less prominent attack on standalone and networked computer systems known as CryptoViral attacks. These are very powerful attacks, where the attacker can hold the victim's data for ransom. The organization of this presentation is as follows. We define the basic terminology and then discuss four different scenario's in which a cryptovirus is used to extort information or/and ransom. Scenario 1 is a cryptoviral extortion protocol performed by holding the victim's data as ransom. Scenario 2 is similar to Scenario 1 except for that the virus writer also demands the victim's encrypted text along with the ransom. Scenario 3 describes a secret sharing cryptovirus. The attack works on a computer network with infected hosts. In this attack the asymmetric private key is split and shared among the infected network hosts. Scenario 4 explores the role of a cryptovirus in a Deniable Password Snatching (DPS) attack commonly used in espionage.

Definitions

Cryptology: *Cryptology* is subdivided into two disciplines, *Cryptography* and *Cryptanalysis*. *Cryptography* is the science of making the secret unintelligible and *cryptanalysis* is the science of retrieving the secret from the unintelligible data.

Virus: A **computer virus** is a program or piece of code written with malicious intent, so to alter the normal working of the computer and is done without the knowledge or permission of the legitimate user.

Worms: A *worm* is a piece of code that uses security flaws to create copies of it. The copy will then look forward to replicate itself by looking for other vulnerable machines. This process continues until all the system resources are consumed thus rendering it useless for any other work.

Trojan: A Trojan is a piece of code that claims to be something desirable but in practice is malicious. Once a user is tricked into running a trojan, it acts as a backdoor to the computer on which it is running. Trojans can be potentially malicious and may cause widespread damage such as erasing files, stealing of passwords and so on.

Cryptovirology: *Cryptovirology* is the term coined to the use and study of cryptology in virus writing in a variety of novel ways for malicious intent namely

- Give malicious software (malware) enhanced privacy and robustness against reverse engineering.
- Give the attacker enhanced anonymity when communicating with the malware.
- Improve the ability to steal data and carry out extortion and so on.

Cryptovirus: A *cryptovirus* is a virus that makes use of a cryptographic public key known to the author of the cryptovirus. *Cryptoworms* and *CryptoTrojans* are same as *CryptoViruses* except that they are worms and trojans.

TRNG: A *TRNG* is a True Random Number Generator. Given any sequence or set of sequences generated by a TRNG, it is impossible to derive any further or previous sequence.

IV: Initialization Vector (*IV*) is a pseudo random number used in a cryptographic mode known as chaining.

CBC: Cipher Block Chain (*CBC*) is a chaining mode that prevents the same plaintext from encrypting to the same ciphertext under the same encryption key.

Threats

Cryptoviral Extortion

Scenario 1

The cryptoviral extortion protocol is performed with the help of a cryptovirus. It is a three round protocol between an attacker and the victim. The attack works using a hybrid encryption scheme that works by encrypting the data as well as deleting and overwriting the actual information on the victim(s) computer.

An asymmetric key pair is generated by the virus author and embeds the public key into the virus. The corresponding private key is with the virus author.

- 1.) The virus author introduces the virus on a public communication medium such as the Internet. More and more hosts get infected on exposure to the virus when no safeguards are in place. Once on the victim(s) machine, the cryptovirus springs into action. It creates a random symmetric key and IV using a TRNG. The data on the secondary storage device is encrypted using this key and chained using a chaining mode as CBC. The actual information is then deleted and overwritten. The IV is appended to the symmetric key and encrypted using the virus writer's public key. The encrypted plaintext is then held at ransom. The encrypted plaintext and anonymous ways to contact the virus author are displayed on the victim(s) screen.
- 2.) If the victim agrees to pay the ransom, he transmits the encrypted (IV, Symmetric key) pair to the virus writer. The virus writer now decrypts the pair using the corresponding private key and sends the (IV, Symmetric key) pair to the victim.

3.) Using the (IV, Symmetric key) pair, the victim is now able to decrypt the information on secondary storage medium.

The attack is ineffective if the data can be recovered from backups. It is however common that many organizations have a daily or weekly backup policy. In such cases the cryptoviral attacks are still valid but over a shorter period of time. The effectiveness of the attack also depends on the sensitivity of the information, the amount of working hours required to duplicate the result and so on.

In practice a fast and compact code algorithm such as the Tiny Encryption Algorithm (TEA) can be used for symmetric key encryption. A hybrid encryption scheme is used here so as to increase the chances of the victim cooperating to pay the ransom. It is to be noted that using the above scheme, the virus writer never comes to know of the content that is held at ransom. This is a model of a scheme from which the thief profits, though he doesn't take anything.

Another interesting question is how the virus attacker will receive anonymous cash payment. Truly anonymous e-cash networks are critical to maintain the virus writer's anonymity during the ransom payment electronically.

Scenario 2

A variant of this attack is described as follows: The attacker in addition to the ransom demands the encrypted text C . It is known as an Information Extortion Attack.

Let

$M = \{ \text{Chksum}, \text{IV}, K \}$

$D = \text{RSA}_{\text{pub}} \{ \text{Chksum}, \text{IV}, K \}$

$\text{RSA}_{\text{pub}} = \text{RSA Public Key}.$

$\text{RSA}_{\text{pri}} = \text{RSA Private Key}.$

$\text{Chksum} = \text{Checksum of the information, wanted by the virus writer}.$

$\text{IV} = \text{Pseudo Random Initialization Vector}.$

$K_s = \text{Random Session Key / Symmetric Key}.$

$W = \text{Virus Writer}.$

The virus writer W demands the ciphertext C and parameter D from the victim. The victim sends the virus writer Ciphertext C and parameter D . The virus writer uses RSA_{pub} to determine the value $M = \{ Chksum, IV, K \}$. Using the key K_s and IV , the virus writer is able to decrypt C and obtain the corresponding plaintext. The checksum of the plaintext is calculated and verified with the $Chksum$ of M . This is to ensure that the victim does not cheat the attacker by providing another text encrypted by the cryptovirus.

Scenario 3

The cryptovirus can be made into a secret sharing virus. The idea works in the following setting:

We have a local network with n hosts. The mode of attack is similar to the previous scenarios except for that the RSA private key RSA_{pri} is now split among k or more nodes, $k < n$ and RSA_{pri} no longer resides with the virus author. The advantage is that the virus writer no longer needs to handle the RSA private keys and the victim need not send anything other than the ransom to the virus writer.

The secret sharing scheme takes advantage of the fact that the RSA private key will be split across k or more hosts on the network making it hard for the system administrator find them and recover them.

Scenario 4

We explore the role of a cryptovirus in a Deniable Password Snatching (DPS) attack commonly used in espionage. An attack is said to be DPS if the attacker is able to recover passwords from the system in a manner that the attacker is untraceable. I.e. even if the malware is discovered the attacker cannot be proven guilty. The other condition for DPS is that the stolen login/password pairs are only accessible by the virus writer and no one else including the administrator of the infected host.

A DPS attack is performed as follows:

The attacker needs to put a CryptoTrojan into the targeted computer. If he directly goes and inserts the cryptotrojan, he has a good chance of being caught in the act. Moreover

evidence collected pointing towards the attack should worsen the attackers chance of being convicted for the crime. Rather, the attacker chooses a passive channel such as Internet Bulletin Boards or infected CD's or diskettes that are swapped without the victim's knowledge. This is to ensure that the cryptotrojan cannot be traced back to the attacker and cannot be held responsible for the attack.

We now look at how the cryptotrojan makes use of the El-Gamal Cryptosystem to make the attack work. We give a brief overview of the El-Gamal encryption algorithm before we describe the attack.

El-Gamal Cryptosystem

- Let g be a generator modulo p , where p is a large strong prime.
- The private key is x where $x < p-1$. The public key is (y,g,p) where $y=g^x \text{ mod } p$.
- To encrypt a message m , we compute $a=g^k \text{ mod } p$. The ciphertext is the pair (a,b) . m must be less than p and k is a randomly chosen number less than $p-1$.
- To decrypt and recover m we compute $ba^{-x} = m.y^k$. $a^{-x} = m.(g^x)^k.(g^k)^{-x} = m$.

The cryptotrojan now creates a file of fixed size on the disk (assuming that there is free space). This data file will be used to store N (login name, password) pairs keyed in by the user. To start with the, cryptotrojan creates $2*N$ random numbers modulo p , where p is a large strong prime. To an outside user this file will look as gibberish.

The Trojan is now ready and waits for the user to key in the username and password. It is assumed that the Trojan has the necessary root privileges and that it is able to infect the system using a known exploit.

- Once a (login name, password)pair is entered, the Trojan encrypts the pair.
- The cryptotrojan using a True Random Number Generator(TRNG) generates a pseudo random positive integer i between 0 and N . The cryptotrojan stores the encrypted pair at the i^{th} entry in the file.
- Before doing this the cryptotrojan generates N pseudo random numbers, namely k_1, k_2, \dots, k_N . and encrypts the (login name, password)pair separately, one at a time

with the N different pseudo random numbers. The encrypted pairs are placed back into the data file sequentially at the locations other than that with entry i .

This is done to cheat the auditing or other monitoring tool. What we have achieved is that we have disassociated the entries in the file. If we were to make a second copy of the data file when another (login name, password) is captured, it would have no similarity to the first.

- This process continues and when the i^{th} (login name, password) is snatched, the i^{th} entry of the data file is overwritten by the corresponding ciphertext pair.

Counter Measures

1. Permanent and direct memory monitoring is necessary to catch self encrypting, polymorphic cryptoviruses.
2. Passwords alone are inherently weak to authenticate to a host or computer system. Two factor authentication, first with the password and the second with a biometric entity such as the users fingerprint, iris scan and so on should be used.
3. Use of up-to-date anti-virus machinery, since cryptoviruses spread the same way as normal viruses does.
4. Access control to cryptographic tools and API's. It is strongly encouraged to audit cryptographic utilities. This is one of the major issues that are commonly overlooked. This will help the system administrator to identify suspicious cryptographic usage.
5. We note that if strong cryptographic routines and good pseudo random number generators are available, it makes the virus smaller and simpler to code and faster to execute if the code is optimal. Incorporating strong cryptographic tools into the operating systems may increase system security but makes it vulnerable since viruses can make calls to these operation system routines. It is hence essential to monitor the processes invoking the cryptographic routines and try to prevent and log processes that do not have sufficient access privileges to call the crypto toolkit/library.

6. Use of Firewall and Intrusion Detection System to protect single and networked systems.
7. Applying patches as soon as it is made available and watching out for zero day exploits.

A few interesting articles can be found in the bibliography. [05] describes the analysis of a cryptovirus by Websense Security Labs. [04] describes the analysis of the Cryzip Ransomware Trojan Analysis *by LURHQ Threat Intelligence Group*. [06] is a description of a real life cryptoviral attack that appeared in the news.

Bibliography

[01]. Adam Young and Moti Yung, *Cryptovirology: Extortion-Based Security Threat and Countermeasures*, Proceedings of the 1996 IEEE Symposium on Security and Privacy.

[02]. Adam Young and Moti Yung, *Deniable password snatching: On the possibility of Evasive Electronic Espionage*, the 1997 IEEE Symposium on Security and Privacy.

[03]. Cryptovirology.com FAQ. Available: <http://www.cryptovirology.com/>

[04]. LURHQ Threat Intelligence Group, *Cryzip Ransomware Trojan Analysis*. Available: <http://www.lurhq.com/cryzip.html>

[05]. Websense Security Labs, *Malicious Website / Malicious Code: Cyber Extortion Attack, May 23, 2005*. Available:

<http://www.websense.com/securitylabs/alerts/alert.php?AlertID=194>

[06]. News by Ryan Naraine, *Cryzip Trojan Encrypts Files, Demands Ransom* March 13, 2006. Available:

<http://www.eweek.com/article2/0,1759,1937408,00.asp?kc=EWRSS03119TX1K000059>