Cyber Espionage

Defending Against The Digital Cloak and Dagger Stealing Information Securely, Privately, and Deniably

Jay Bavisi, EC-Council

Abstract. As multi-disciplined approaches to offensive computing take shape, combining everything from cryptographic protocols to anonymous communication channels to malicious software, we are starting down a logical path of evolution where various methodologies converge into something that often functions outside the scope of the individual methods themselves, hitting us with attack vectors we barely understand, let alone for which we have a solid defense.

In the case of cyber espionage, covert offense is the underlying goal of an adversary. The subtlety of the attack is paramount since drawing too much attention may alert the victim and result in immediate countermeasures, thus revealing, and halting, any attack efforts.

To ensure the secrecy and privacy of these activities, an adversary might deploy malicious software that is capable of stealing information without revealing what it is that the adversary is looking for or what the adversary has taken–which is known in cryptography as Private Information Retrieval (PIR).

First, let us look at the basic algorithmic model of a PIR-based attack, based on the foundation that an adversary is attempting to privately retrieve information from a database without the database administrator's knowledge. Suppose we have a database consisting of n entries where each entry holds a single binary bit; the database can be represented by a bit string, $B = b_1 b_2 \ldots b_n$.

The adversary needs to submit a query (i) to the database administrator, where $1 \leq i \leq n$, such that i is not revealed to the database administrator, while b_i is returned as a response to that query. In this manner, b_i has been privately retrieved from the database by the adversary.

Consider the following simplified attack model within the context of corporate espionage: An adversary–let us say an actual employee, an insider–is disgruntled by the possibility that another employee is being overly compensated. The adversary designs a piece of malicious software, in the form of a virus that contains a tag string. Depending on how the database is indexed, this tag string could be programmed to look for the salary information of a given employee, for example.

With that in mind, when the virus matches its tag string to an entry in the database, it knows it has found its target and internally stores the data within itself. Because this data is stored clearly and could potentially be discovered by anyone who obtains the virus, the adversary would encrypt the data using a public key, for which he possesses a corresponding private key that will be used to decrypt the data when the adversary obtains the virus. This is referred to as a cryptovirus, which can even make use of the cryptographic API (e.g., Microsoft's CryptoAPI) within the victim's host system, thus allowing the cryptovirus to be even more lightweight, by eradicating the need for an onboard cryptosystem.

In addition, because snapshots of a virus may reveal the tag string, which could also be obtained by anyone, the aforementioned PIR algorithm is used to prevent this from occurring by concealing the tag string after the adversary retrieves the data of interest. Of course, we can imagine applications outside of corporate espionage–governments, militaries, financial institutions, etc.–and this is just a method for stealing specific, isolated data What if we could privately steal passwords and covertly obtain access to hordes of information? We can.

Take the notion of a cryptovirus and mesh it with the notion of a Trojan horse; now you have cryptotrojan. To further the idea of stealing information, it would be even more ideal if the adversary had some sense of deniability, such that even if his activity is noticed, it cannot be proven beyond a shadow of doubt that he is actually attacking–despite the use of auditing or logging mechanisms–thanks to the combination of cryptography and mix network channels that provide anonymity.

Not only that, but the cryptotrojan can be designed in such a way as to prevent the victim, upon possible discovery of the cryptotrojan, from determining which passwords, if any, have been stolen or even whether or not the cryptotrojan has stolen anything at all. The implications of this are substantial, as we have now combined malicious software, communication channels, and cryptographic algorithms to build a platform for deniable espionage–cyber spies that can not be caught, even if they are caught.

We are already seeing, in the real world, evidence of the adversaries' consciousness of malicious software that uses cryptography to cover itself with advanced worms such as Conficker, implementing state-of-the-art cryptographic algorithms to prevent the hijacking of payloads. And, with other recent large-scale attacks, as we have seen with the Stuxnet worm and the espionage network, GhostNet, it is not hard to hypothesize the significant impact it would have should the malware of the future employ these covert techniques to evasively wreak havoc, and leave us scratching our hands as to what just happened, and who we are to thank for it.

Detecting Abnormal and Cryptographic Code

Working from the common knowledge that a virus often appends itself to an object such as an executable, and changes the point of entry to itself, you now have a heuristic by which to detect it. Certain detection techniques operate on the premise that custom malicious software will produce statistical discrepancies when they manifest themselves through otherwise ordinary binary executables; these techniques seek out such anomalies and facilitate the detection of potentially malicious code.

Heuristic analysis can be used to pinpoint probable malicious code the same way it can also be used to pinpoint the existence of cryptographic code. Previously, we discussed the use of pub-

lic and private cryptographic keys to conceal the data pilfered by the cryptovirus. Because common asymmetric cryptosystems are based on the integer factorization problem, which is the issue of determining the prime factorization of a given integer, cryptographic APIs will often test primality by performing trial division on small primes and store a list of the results which could potentially be discovered using string matching.

Taking this a step further, because asymmetric cryptography's arithmetic-intensive nature can render many implementations inefficient, an algorithm known as the Karatsuba algorithm can be used to speed up the multiplication between two large numbers. Along with detecting the presence of such efficiency-boosting algorithms for key generation, the actual presence of public keys themselves can be investigated in long bit strings via specific algebraic methods, or in large programs, via more general statistical methods. Because cryptographic keys depend on randomness, they will exhibit much more entropy than surrounding data, which is more structured and redundant. In a visual sense, cryptographic keys would appear to be noisier than surrounding data.

Unfortunately, this brings to light the double-edged nature of cryptography, which is typically used as a defense mechanism for preserving goals like confidentiality and integrity. On top of that, cryptography is arguably the strongest link in any system. This strength translates directly into the malicious use of cryptography, making intrusion detection and response particularly difficult.

Spies of old, despite their tools and savvy, would, if caught, fear for their own life–be it spent in prison or swiftly ended by other means. Denial does not work too well. For digital spies however, it can, and thanks to the nefarious use of our own weapons against us, there is not always a solution. This frontier is relatively new and sparsely explored, and judging by the caliber of what is possible now, it is likely that we will be even more impressed, or depressed, by what lies ahead, depending on which side you are on.

ABOUT THE AUTHOR

Jay Bavisi is the co-founder and president of EC-Council, the governing body of the world-renowned Certified Ethical Hacker CEH program. A distinguished author and speaker on information security, he has given keynote speeches at international conferences, been invited to lecture at international corporations and academic institutions, contributed numerous articles to major technical publications, and had his views sought after by internationally acclaimed media giants, such as The Wall Street Journal, CNN, TIME Magazine, and USA Today. Mr. Bavisi is a law graduate from the University of Wales, College of Cardiff, having an LLB (Hons), Barrister–at–Law from Middle Temple, London.