+1
1 Votes

Honeytokens: The Other Honeypot

By: ()

Created 16 Jul 2003 | 1 Comment

0 4

by Lance Spitzner

Authors Note: This paper was originally released on 17 July, 2003. In the days following its release, I received more feedback on this paper than on all other previous honeypot-related papers combined. As a result, I decided to release an updated version that broadens the scope of honeytokens, based on the tremendous input of the community. I hope you find this updated version useful.

The purpose of this series (http://www.securityfocus.com/infocus/1659) of honeypot papers is to cover the breadth of honeypot (http://www.tracking-hackers.com/papers/honeypots.html) technologies, values and issues. I hope by now readers are beginning to understand that honeypots are an incredibly powerful and flexible technology. They have multiple applications to security, everything from simplified detection to advanced information gathering. Today we extend the capabilities of honeypots even further by discussing honeytokens. Honeytokens are everything a honeypot is, except they are not a computer.

Definition

One of the greatest misconceptions of honeypots is they have to be a computer, some physical resource for the attacker to interact with. While this is the traditional manifestation of honeypots, its not the only one. Take into consideration the definition of the honeypot, as defined by the honeypot mailing list (http://www.securityfocus.com/popups/forums/honeypots/faq.shtml).

"A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource."

Note in the definition that we do not state a honeypot has to be a computer, merely that its a resource that you want the bad guys to interact with. That is exactly what a honeytoken is, a honeypot that is not a computer. Instead it is some type of digital entity. A honeytoken can be a credit card number, Excel spreadsheet, PowerPoint presentation, a database entry, or even a bogus login. Honeytokens come in many shapes or sizes, however they all share the same concept: a digital or information system resource whose value lies in the unauthorized use of that resource. Just as a honeypot computer has no authorized value, no honeytoken has any authorized use. Whatever you create as a honeytoken, no one should be using or accessing it. This gives honeytokens the same power and advantages (http://www.securityfocus.com/infocus/1690) as traditional honeypots, but extend their capabilities beyond just physical computers.

The concept of honeytokens are not new. This concept is as old as security itself. For example, I've read discussions of map-making companies inserting bogus cities or roads into their maps to determine if competitors are selling copied versions of their own maps. In Cliff Stoll's book "The Cuckoo's Egg," we learn how he deploys digital files to track and monitor a German hacker. While the concept is not new, the term is. What makes honeytokens new and exciting is we now have a single term to bring these concepts together for the information security community.

The term *honeytoken* was first coined by Augusto Paes de Barros in 2003 on the honeypots mailing list. This term aptly described the concept, and as is often true when dealing with technologies, having a commonly accepted term makes the concept easier for others to understand and discuss. That is what this paper attempts to do.

How They Work

A honeytoken is just like a honeypot, you put it out there and no one should interact with it. Any interaction with a honeytoken most likely represents unauthorized or malicious activity. What you use as a honeytoken, and how you use it, is up to you. A classic example of how a honeytoken could work is the "John F. Kennedy" medical records example. Under HIPAA, hospitals are required to enforce patient privacy and only certain authorized people have access to patient data (such as doctors, nurses, etc). If a hospital fails to protect patient data, that hospital (and the individuals involved) not only face civil liability, but possibly criminal liability. If a hospital has people looking where they shouldn't they want to know immediately. A honeytoken could be used for such a purpose. Traditionally, this has been very difficult to do. How do you detect when you have unauthorized access to a database when that database has thousands, if not millions of records, with hundreds of authorized users? Maintaining who is authorized to what can be complex, with false positives becoming a huge problem. Honeytokens can be used to solve and simplify this problem.

A bogus medical record called "John F. Kennedy" is created and loaded into the database. This medical record has no true value because there is no real patient with that name. Instead, the record is a honeytoken, an entity that has no authorized use. If any employee is looking for interesting patient data, this record will definitely stand out. If the employee attempts to access this record, you most likely have an employee violating patient privacy. It is as simple as that, no fancy algorithms, no signatures to update, no rules to configure. You load the records, monitor it, and if someone accesses it they most likely have violated the system's usage policy.

Another example is bogus Social Security or credit card numbers. We have read numerous stories of large databases compromised, with thousands of SSNs or millions of credit card numbers compromised. Even worse, often these compromises are not detected for weeks if not months later. This gives attackers extensive amounts of time to use or sell the information. Honeytokens can once again be used to simplify this problem. Bogus numbers can be embedded in a database. If the numbers are accessed, you know someone is violating system security. A university could put SSN honeytokens in their student database. If someone attempted to steal the entire database (as has happened at several universities) the attackers would also be grabbing the honeytokens mixed with the valid SSNs. The same could be done for credit card numbers embedded into a vendors on-line ecommerce site. These honeytokens would be unique numbers, so attackers would not know what was the honeytoken and what were valid numbers. Databases could watch for whenever someone attempted to access the records and generate an alert. Or, IDS sensors could be configured to watch the local networks. If these honeytoken numbers are detected on the wire, then the databases have most likely been compromised.

For example, the credit card number 4356974837584710 could be embedded into database, file server, or some other type of repository. The number is unique enough that there will be minimal, if any, false positives. An IDS signature, such as Snort, could be used to detect when that honeytoken is accessed. Such a simple signature could look as follows.

```
alert ip any any -> any any (msg:"Honeytoken Access - Potential Unauthorized Activity";   content:"4356974837584710";)
```

This concept can easily be extended beyond databases. File, web, or email servers can all have honeytokens embedded into them. Anything that has data can easily have additional bogus data added, bogus data that becomes our honeytoken. File servers could have bogus files, such as Word documents, .pdf files, or Excel spreadsheets. These files could have unique names, or unique tags embedded in the files. Intrusion Detection Systems can then have signatures customized to look for these honeytokens. If you see any honeytokens traversing your networks, you know you have employees (or someone on your internal network) accessing files they are not authorized to. For encrypted environments, kernel modules can be developed to monitor system files. If someone attempts to read() one of the files residing on the system, the kernel module can detect this unauthorized activity and generate an alert (the Honeynet Project (http://www.honeynet.org) has such concepts currently under development).

Value

Just like traditional honeypots, honeytokens do not solve a specific problem. They are not designed specifically to detect blackhats or prevent attacks. Instead, they are a highly flexible and simple tool with multiple applications to security, everything from detection to identifying who your threat is and their motives. Honeytoken's value lies in their simplicity. You deploy a digital file or record and if anyone accesses them, you potentially have a problem. I see honeytokens used primarily for the insider threat, for the trusted individuals on the inside. You leverage the fact that the insider attacker knows your internal environment and has access to files, information and records (including our honeytokens) that untrusted outsiders would not have access to. Also, I feel honeytokens should not used by themselves. As true with almost any technology, their real value is when they are combined with other solutions. For example, in many cases honeytokens may not prove unauthorized activity. Instead, they may merely indicate you have unauthorized behavior. You most likely will have to use other tools to confirm if someone is acting with malicious intent. An employee may access a honeytoken that is a Microsoft Word file posing as your company's Research and Development plans. If an employee attempts to copy and transfer the file, you have identified a problem. Either you have a confused employee looking where they shouldn't, or you have someone who is knowingly attempting to access high-value documents they do not have authorization for. Regardless, once you have identified this activity, you can then use other measures to confirm the individuals intent.

In my mind, what exponentially increases honeytokens value is the cost involved, which is minimal. Think about it, there is no technology to deploy, no vendors to contact, no licenses to update. What do you need to develop a honeytoken? At most, you create a unique PowerPoint file, an image, or bogus record. Compared to many other technologies, honeytokens represent one of the simplest and most cost effective security tools.

Finally, honeytokens have extensive flexibility. You are limited only by your imagination. As we have demonstrated in the section above, honeytokens excel as a detection mechanism. However, honeytokens can do so much more. Not only can they detect an attacker, but they can potentially identify who that attacker is and what they are after. Let's say a company is concerned about internal employees attempting to find company secrets. Honeytokens can be used to smoke them out and identify who they are. How you do that is simply a matter of how creative you want to get. For example, in our case we could plant honeytokens in senior management's email. The plan being, an internal employee may be reading management's email to gain access to privileged information. To track such unauthorized activity we create a bogus email, or honeytoken, and plant that in management's email. The email could look like this:

```
To: Chief Financial Officer From: Security help desk Subject: Access to financial database Sir, The security team has updated your access to the company's financial records.  Your new login and password to the system can be found below. If you need any help or assistance, do not hesitate to contact us.     https://finances.ourcompany.com (https://finances.ourcompany.com)     login:    cfo     password: H0n3yt0k3n Security Help Desk
```

If an attacker is cruising through emails and comes across this, they most likely will attempt to access the financial server thinking they can retrieve highly confidential data. What they don't know is they just hit a honeytoken. The system 'finances.ourcompany.com' is really a honeypot watching the network for unauthorized activity. The moment someone connects to the honeypot, you have identified a problem. The moment someone connects to the honeypot and uses one of your honeytoken logins, you know you have someone reading senior management's email. The moment such a connection happens, you immediately initiate a trace back to identify the computer the attacker is on and potentially who the attacker is. For distributed use of honeytokens, you can plant a different login and password in each email for each VP, thus tracing back exactly whose email has been compromised. In addition, within the honeypot you can place different bogus records (additional honeytokens). These can be files with different names or data, such as sales figures, salaries, budgets, banking records, etc. You can then not only identify who the attacker is, but what the attacker is after by which records they pull. In "The Cuckoo's Egg," Cliff Stoll did exactly this, planting files to determine the attacker's motivation. What was even more clever, Cliff took it a step farther by putting a mailling address in the documents. The attacker was lead to believe that by mailling requests to the address, he could gain access to more documents. Not only now was Cliff able to detect the attacker's motives, but now he could track the attacker down, as the attacker was unknowingly communicating directly with him. Who needs to build a "call home" functionality into the honeytoken when the attacker does it for you? The options are almost unlimited on how this technology can be applied.

Conclusion

Honeytokens are an exciting new dimension for honeypots, especially for the insider threat. They are cost effective, simple to deploy, and highly effective. Honeytokens represent an entirely new field for honeypot concepts, and expect to see much more development in this area. Next month we cover a deployment concept for honeypots: the honeypot farm. Instead of deploying honeypots all over your networks, you install all your honeypots in one place, then let the attackers come to you.

Related Articles
View more articles (http://www.securityfocus.com/search?category=13&query=lance+spitzner) by Lance Spitzner (mailto:lance@tracking-hackers.com?Subject=Honeytoken Paper) on SecurityFocus.

Tags: Security (/connect/search?filters=im_vid_31:691), Endpoint Protection (/connect/product/endpoint-protection-vdi), SecurityFocus (/connect/search?filters=im_vid_4:4521)

Subscriptions (0)

()

()

View Profile ()

Jump to latest comment

1 Comment

(/connect/user/simoneng)
simoneng (/connect/user/simoneng)

Hi,

Honeytoken is a very neat concept. Is there out-of-box support for honeytoken in Symantec's DLP (Vontu)? If yes, perhaps can share the URL of website where we can get more info?

Thanks in advance.

Simon

Simon Eng
"Question to learn, Learn to Question"

Login (/connect/user/login?destination=node%2F1234461) or Register (/connect/user/register?destination=node%2F1234461) to post comments.
01 Mar 2012 (/connect/articles/honeytokens-other-honeypot#comment-6790491)

Login (/connect/user/login?destination=node%2F1234461) or Register (/connect/user/register?destination=node%2F1234461) to post comments.