Vulnerability Disclosure Report

SIMPLEIT® SECURITY SERVICES

IT management solved, *IT's that simple®*

Discovery Date: December, 2017
Reported to Vendor Date: January, 2018
Public Notification Date: January, 2019

simpleroute

simpleroute
1690 Williston Road
South Burlington, Vermont
05403

802-881-0010
security@simpleroute.com
simpleroute.com

New England Municipal Resource Center (NEMRC) Software Vulnerabilities Disclosure

For immediate release

Abstract

Systems that hold public information are critical targets for attackers. Most municipalities in Vermont rely on a piece of software called NEMRC made by the New England Municipal Resource Center (NEMRC) for holding town and resident data. This paper covers the discovery of three vulnerabilities in the NEMRC software found by simpleroute and reported to NEMRC in January 2018. simpleroute is a Managed Service Provider located South Burlington, Vermont that provides IT services, security and cloud services to small and medium sized businesses. At the time of publication, each of these vulnerabilities have been patched. Users not current on the software at risk of the deficiencies noted in this report. Two of the patched vulnerabilities may have resulted in scenarios that would allow unintended exposure of municipal employees' social security numbers (SSNs) in addition to municipal tax payer banking information including routing and bank numbers. Evidence obtained from our client backups show these vulnerabilities may have existed in software as far back as February 2006. The third vulnerability was recently fixed and affects the integrity of data sent to NEMRC's cloud. Substantially more than the industry average 90-day window of time prior to public disclosure has been provided to allow affected parties to update affected systems. It is hoped that this disclosure will provide critically necessary information for municipalities to protect themselves and encourage public debate on the security of municipal IT information systems and the security services deployed across municipalities.

Introduction

Municipalities require some form of electronic record keeping system to hold vital data. This data can include everything from marriage licenses to tax payments plus human resource and asset information. While this data was once present on paper, much of it is now electronic for cost and ease of use reasons. However, this presents a growing problem as electronic records require diligence and security controls to ensure non-public data held by municipalities are safely kept private.

The move to and reliance on the NEMRC application throughout municipalities in the state of Vermont has become pervasive. NEMRC publicly notes it is in use at roughly 200 municipalities across Vermont in addition to being in use by the State of Vermont, State of Maine and roughly 100 other scattered municipalities [1].

Information security requires a layered approach. Application vendors bear but one piece of this level of responsibility. Some of the security layers that can help prevent data breaches are as noted below:

a) Perimeter security
b) System security
c) Domain security
d) Share security
e) File permission security
f) Database security
g) Application security

Each of these layers play a role in protecting application data. While application creators may only control or influence a few of these layers, municipalities have historically struggled to protect information assets as these protections require adequate budgets that can be difficult to justify in the face of a lack of public understanding of the issues and potential challenges at hand.

Through simpleroute's diligence in verifying vendor applications and analyzing IT security, we have uncovered several security issues with the NEMRC software. It is known that these issues existed for a significant window of time before discovery and may have been present as far back as February 2006 based on analysis of client backup data. Each of the identified issues and their potential impacts are detailed below in a separate section.

NEMRC operates cloud and on-premise versions of their software. Currently, it is unknown to simpleroute what the client breakdown between these models is. All vulnerabilities were discovered and tested on the on-premise version of the software. The affectedness of the cloud version of the product is presently unknown.

Background

The on-premise NEMRC software is a Visual FoxPro 7 backed application which contains a flat file backend with a client-side application frontend. Any user of the NEMRC application requires full access to the NEMRC files on the server for the client-side frontend to operate properly.

This setup means that improper security on the database tables could lead to unintended data access by any party that uses the NEMRC application as all users require full access to the backend flat files. It also means any files placed in the server share used by the NEMRC client application would be accessible to any user who used or was granted access to NEMRC.

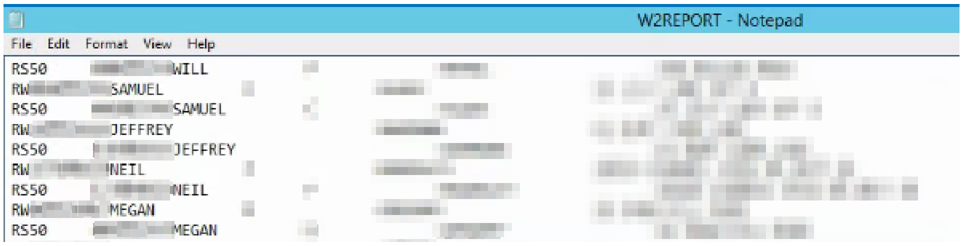
During our discovery, we found that many municipalities historically granted all town staff access to the NEMRC share simply due to lack of data security controls or a lack of understanding of the principle of least privilege. Entities that did not properly restrict sharing permissions on the NEMRC share and relied solely on the NEMRC application security would in turn be at risk for a higher level of exposure of the vulnerabilities contained later in this document.

On-premise clients are also provided an additional option of subscribing to cloud backup services provided by NEMRC for the NEMRC application files. For users who opt-in to this service, the NEMRC application will upload NEMRC database files regularly to a server maintained by NEMRC. Many on-premise clients subscribe to this option as an additional safeguard to other backup protections they may have in place.

This document is broken down into three sections that review each of the vulnerabilities discovered in the NEMRC application by our team.

1. Attack Overview: Potential Municipal Employee Social Security Exposure

simpleroute noted that a file called W2REPORT was in the root of the NEMRC files on servers running the NEMRC software. Analysis by simpleroute uncovered unencrypted SSNs for municipal workers contained within this file. In fact, this file was a plaintext copy of the last W2 report municipalities generated for upload to the IRS. We have enclosed a heavily redacted screenshot of this data below (the first scrambled entry in each row contains the municipal worker's SSN).



```
RS50 [redacted]WILL [redacted]
RW[redacted]SAMUEL [redacted]
RS50 [redacted]SAMUEL [redacted]
RW[redacted]JEFFREY [redacted]
RS50 [redacted]JEFFREY [redacted]
RW[redacted]NEIL [redacted]
RS50 [redacted]NEIL [redacted]
RW[redacted]MEGAN [redacted]
RS50 [redacted]MEGAN [redacted]
```

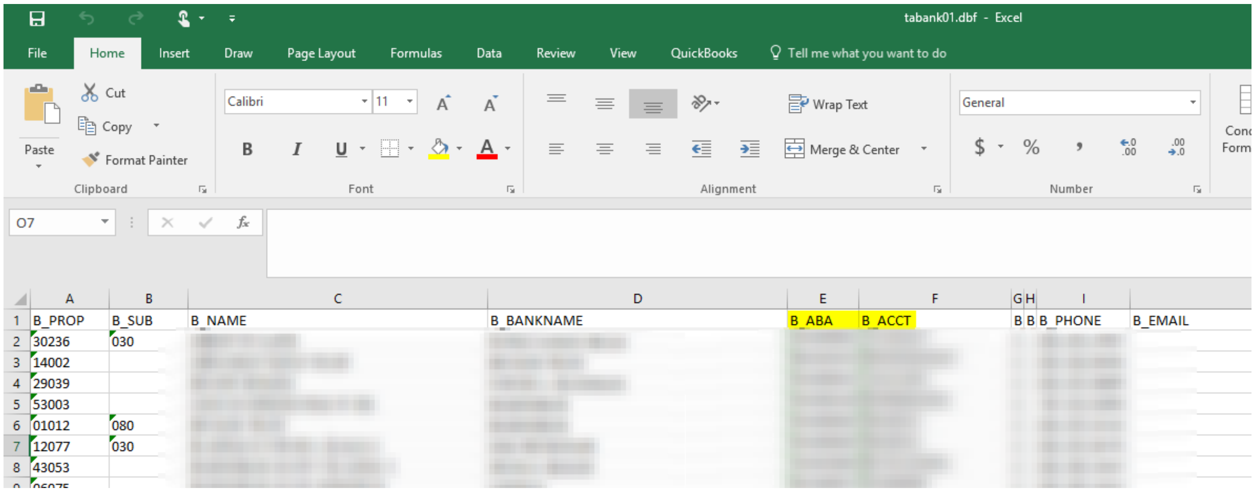
While this file could be deleted, it was found by simpleroute that it would be recreated anytime the W2 report was later run. In affected Towns managed by simpleroute, finance users were not aware the W2REPORT file was being saved to the root of the NEMRC share. In running the NEMRC client and generating a W2 report, the application would prompt the user to save the report. While most users would pick a safe/secure location to save this data, no mention was made by the NEMRC application that data was being stored in the root NEMRC share in plaintext with visibility to all NEMRC users.

2. Attack Overview: Taxpayer Routing and Bank Information

As NEMRC relies on Visual FoxPro 7, it's database tables can be easily opened, and the contents viewed with many applications including Microsoft® Excel. As previously noted, any user with NEMRC share would have sufficient privileges to view data in the underlying database tables. While NEMRC encrypted data in most tables, this appears to have been a manual process by programmers as it was not done at the database layer and only selective columns were encrypted.

Analysis by simpleroute noted that one file, tabank##.dbf (where ## is generally a two-digit number), contained several pieces of sensitive information that were not encrypted. Without any form of NEMRC client access, our engineers were able to uncover the banking accounts, routing numbers, full names and addresses of tax payers entered into the NEMRC software by file access alone. Like the first attack, any user with privileges to see the NEMRC share would have sufficient privileges to see these banking details – even if they had no access to the NEMRC application itself.

The below screenshot shows a heavily redacted version of one such affected client. The B_NAME column corresponds to the tax paying entity, B_BANKNAME to their bank's name and B_ABA and B_ACCT to their routing and bank account data respectively.



	A	B	C	D	E	F	G	H	I	
1	B_PROP	B_SUB	B_NAME	B_BANKNAME	B_ABA	B_ACCT	B	B	B_PHONE	B_EMAIL
2	30236	030	[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	
3	14002		[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	
4	29039		[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	
5	53003		[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	
6	01012	080	[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	
7	12077	030	[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	
8	43053		[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	
9	06075		[redacted]	[redacted]	[redacted]	[redacted]			[redacted]	

Absent manual intervention, this data could have been viewed by anyone with NEMRC share access – even if they were denied from viewing this data within the NEMRC application itself. It's noteworthy that without database level security of this data, overexposure of share permissions could have allowed a significantly larger audience to view this data than intended. Analysis of client backups by simpleroute show the tabank##.dbf file appears in the software dating back to backups dated February 2006. While our clients did not start entering data in these fields when first introduced, our first confirmed occurrence of plaintext data in these fields dates to December 2006.

3. Attack Overview: Insecure Cloud Backup Transport

Investigation into the cloud backup service provided by NEMRC uncovered issues that when combined created a potential attack avenue against entities subscribed to NEMRC's cloud backup service.

When discovered by simpleroute, the NEMRC cloud backup service transmitted backups via File Transfer Protocol (FTP) to a remote server. The FTP transport mechanism itself did not offer any level of encryption. Data sent over FTP, including the FTP username/password could in turn be scraped through a packet capture or man-in-the-middle attack.

While NEMRC did originally take steps to secure the data sent via FTP by wrapping it into an encrypted zip file, captured zip files were shown to be encrypted using a very outdated PKZIP encryption algorithm. The form of encryption used is known as ZipCrypto Store which was developed around 1990. ZipCrypto is an independent algorithm separate from more modern and secure methods such as 3DES, AES and RSA. The algorithm used has publicly posted vulnerabilities which were published as far back as 1994. Researchers Eli Biham and Paul Kocher publicly noted the following regarding use of this algorithm in a 1995 published article [2] on the same encryption algorithm used by NEMRC:

> *"… we describe a known plaintext attack on this cipher, which can find the internal representation of the key within a few hours on a personal computer using a few hundred bytes of known plaintext. In many cases, the actual user keys can also be found from the internal representation. We conclude that the PKZIP cipher is weak, and should not be used to protect valuable data."*

Using this information, the FTP backup of the NEMRC application could be captured in plaintext. The zip file payload could then be scraped from this plaintext communication. The zip file could theoretically be cracked in a mere matter of hours without knowledge of the encryption password used by the NEMRC application.

To test this theory, simpleroute engineers did a packet capture of one such upload and captured a copy of the zip file in transit to the NEMRC datacenter. Engineers were then able to compile a 2003 version of pkcrack 1.2.2 which exploits the weak ZipCrypto algorithm. Feeding nothing more than an empty header file for a Visual FoxPro 7 database into the pkcrack application, we were then able to obtain a key that would open our captured zip file upload and extract any information contained therein – despite our engineers having no knowledge of the zip file encryption password. Example output of our scripted attempt to open the PKZIP file is shown below:

```
./pkcrack -C ~/temp/CAPTURED_FTP_BACKUP.zip -c "tactrl20.fpt" -P
~/temp/blank.zip -p "steril.fpt" -d out.zip -a
Files read. Starting stage 1 on Wed Nov  7 13:45:51 2018
Generating 1st generation of possible key2_26 values...done.
Found 4194304 possible key2-values.
Now we're trying to reduce these...
Done. Left with 2038068 possible Values. bestOffset is 24.
Stage 1 completed. Starting stage 2 on Wed Nov  7 13:45:52 2018
Ta-daaaaa! key0=60959146, key1=e1bfd327, key2=4e99a712
Probabilistic test succeeded for 7 bytes.
Ta-daaaaa! key0=60959146, key1=e1bfd327, key2=4e99a712
Probabilistic test succeeded for 7 bytes.
```

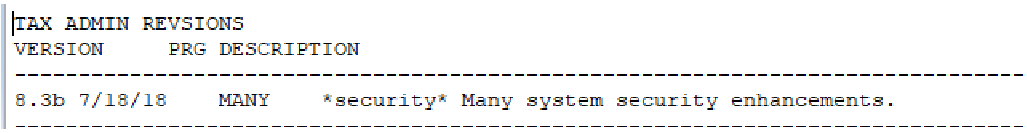
```
Stage 2 completed. Starting zipdecrypt on Wed Nov  7 23:01:12 2018
Decrypting tabank20.dbf (e040ce2371ee15605b43a659)... OK!
Decrypting tabank20.fpt (e040ce2371ee15605b4331ce)... OK!
Decrypting TACASH20.DBF (e040ce2371ee15605b43cf30)... OK!
Decrypting tactrl20.dbf (e040ce2371ee15605b43f906)... OK!
Decrypting tactrl20.fpt (e040ce2371ee15605b4330cf)... OK!
Decrypting tamast20.dbf (e09494657143fa4a0249cc33)... OK!
Decrypting tamast20.fpt (e09494657143fa4a02491be4)... OK!
Decrypting tapass20.dbf (e09494657143fa4a024933cc)... OK!
Decrypting tapass20.fpt (e09494657143fa4a0249d629)... OK!
Decrypting tasyst20.dbf (e09494657143fa4a02493fc0)... OK!
Decrypting tasyst20.fpt (e09494657143fa4a0249ff00)... OK!
Decrypting tatrax20.DBF (e07c4f65bcc3698a0a167887)... OK!
Decrypting tatrax20.fpt (e07c4f65bcc3698a0a16718e)... OK!
Decrypting taX2AP20.dbf (e07c4f65bcc3698a0a16807f)... OK!
Decrypting taX2GL20.dbf (e07c4f65bcc3698a0a164ab5)... OK!
Finished on Wed Nov  7 23:01:13 2018
```

Cracking this took a trivial amount of time – under 10 hours on a single 2-year old Core i5 processor.

Mitigation Options

Due to the scope of entities potentially affected and the type of information at risk, simpleroute provided substantial time for NEMRC to address these issues in their software and then in turn for end-users to apply this update. All users should immediately update to the latest version of NEMRC to fully mitigate these issues.

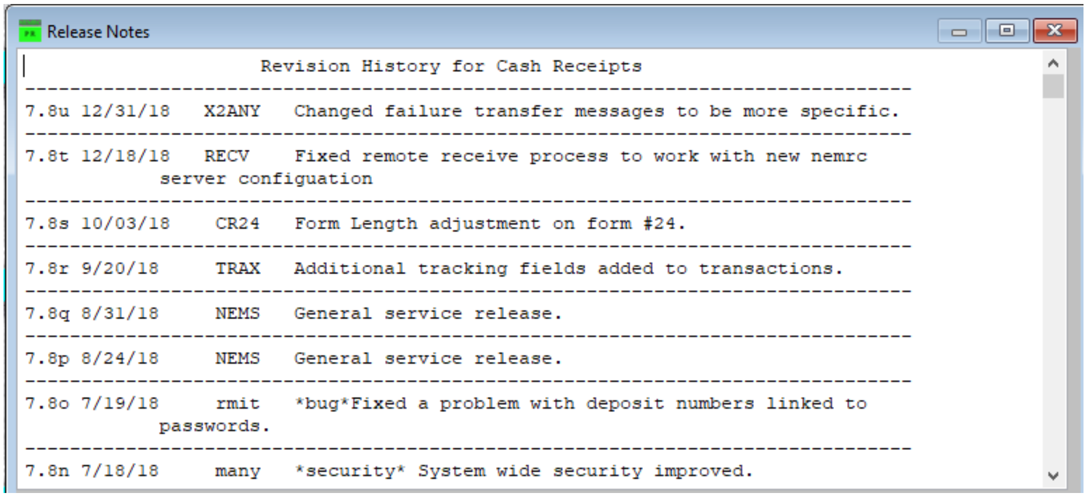
The first two vulnerabilities were confirmed fixed in the July 18, 2018 release (noted as tax admin revisions version 8.3b) per testing by simpleroute. This update contained the below notes upon release:



```
TAX ADMIN REVSIONS
VERSION     PRG DESCRIPTION
-------------------------------------------------------------------------------
8.3b 7/18/18    MANY    *security* Many system security enhancements.
-------------------------------------------------------------------------------
```

simpleroute tested affected clients before/after the patch and noticed issues #1 and #2 were remediated after application of the above patch.

The FTP transport issue was fixed in the December 18, 2018 release (release notes show this as 7.8t). The below release notes correspond to the FTP client upload change:



```
                     Revision History for Cash Receipts
-------------------------------------------------------------------------------
7.8u 12/31/18   X2ANY   Changed failure transfer messages to be more specific.
-------------------------------------------------------------------------------
7.8t 12/18/18   RECV    Fixed remote receive process to work with new nemrc
            server configuation
-------------------------------------------------------------------------------
7.8s 10/03/18    CR24   Form Length adjustment on form #24.
-------------------------------------------------------------------------------
7.8r 9/20/18     TRAX   Additional tracking fields added to transactions.
-------------------------------------------------------------------------------
7.8q 8/31/18     NEMS   General service release.
-------------------------------------------------------------------------------
7.8p 8/24/18     NEMS   General service release.
-------------------------------------------------------------------------------
7.8o 7/19/18     rmit   *bug*Fixed a problem with deposit numbers linked to
            passwords.
-------------------------------------------------------------------------------
7.8n 7/18/18     many   *security* System wide security improved.
```

Updated executables for the NEMRC backup software that resolve this issue were seen with the following size/hash information

```
Name: NEMBACK.EXE
Size: 306824 bytes (0 MB)
SHA256: A965EC231F7D9713F6BC779D9BA0BC22CCD507F7DEF12EE528F38251D80DA90C
```

If you are unsure of whether your installation is fully patched, we strongly recommend contacting a qualified IT professional or organization to advise you and verify you are on the current release.

Proper security practices would also help mitigate this and other issues in the future. The below recommendations are good guidelines for all municipalities to help avoid similar scenarios or potential exposure to similar issues:

a) Perform a security assessment from an independent vendor-neutral IT firm not affiliated with your software vendors. This assessment should include file-level scanning for personally identifiable information (PII).
b) Practice the principle of least privilege and provide only the access necessary for end-users to perform their jobs. Those without need to access a program like NEMRC should not be granted rights to the NEMRC files. Additionally, the NEMRC share should *never* be left with FULL, EVERYONE or ANONYMOUS share/file permissions.
c) Maintain adequate antivirus, antimalware and perimeter security defenses.
d) Perform regular scanning, monitoring and auditing of systems via qualified security personnel.
e) Carefully review software chosen to hold, store and transmit municipal data to ensure that it meets current requirements for data security. As a rule, all data should be encrypted in transit and at rest.
f) Ensure all vendor software is up to date and current. Software either without vendor support or containing components which do not carry vendor support may pose a risk to operations. Current and supported software provides a level of accountability regarding data security.

Conclusions

As information has gone digital, the need to protect it has become paramount. The ability of a thief to gain entry to a municipal building and haul away a truckload of filing cabinets in the night is difficult if not impossible to pull off. However, the ability of a malicious third party to gain access to and copy digital data remains trivially easy without the proper protections in place.

The reliance on a database engine that was end-of-lifed in 2008 [3] in critical municipal software requires that municipalities take extra precautions to ensure that data is not at risk. As the database software is no longer maintained by the upstream software vendor, tertiary means of securing data are required as security vulnerabilities are no longer being addressed by the database vendor.

While simpleroute is not currently aware of any other exploitable flaws in the NEMRC design, it is likely prudent for all parties to move away from Visual FoxPro 7 in the future. In the interim, mitigation techniques are necessary to properly secure assets relying on legacy database software.

We also wish to point out what we perceive to be a shortcoming in the current breach notification laws in place to protect employees and consumers today. At present, there appears to be no requirement to provide notification if proof of a breach does not exist. While the municipalities we provide IT for have no evidence of a breach, the scope of impact is large enough and over a long enough period that it is impossible to say with absolute certainty that municipal data has not been compromised anywhere due to the vulnerabilities noted herein. It is worth public debate surrounding how the public becomes aware of such issues as it is clear potential exposure merits some level of protection and disclosure, even if not at the level of breach notification.

Acknowledgements

Several employees and municipalities contributed to our findings and to helping bring this information safely to light. Particularly, we’d like to thank Daniel Safford for his diligence in performing the initial analysis requested by our company President, Brett Johnson who requested the initial review given concerns he had regarding potential data exposure. We’d like to thank Charles Bucchioni for his work in verifying the zip file transport issue and compiling the legacy exploit code that enabled us to verify the insecurity of the FTP transport upload. We’d also like to thank Mateo Tudon for his involvement in reviewing and spot checking our findings and helping proof materials for release.

While the municipalities that have contributed wish to remain unnamed, we are extremely grateful for their patience, time and trust in helping us unearth these issues and making them known. We’d finally like to thank our state and town municipal employees for their dedication and hard work. You are part of what make Vermont great and we truly appreciate all that you do.

References

[1] New England Municipal Resource Center, "NEMRC | New England Municipal Resource Center," 09 08 2016. [Online]. Available: https://www.nemrc.com/users.php. [Accessed 19 01 2019].

[2] E. Biham and P. C. Kocher, "A known plaintext attack on the PKZIP stream cipher," *Fast Software Encryption*, 02 06 2005.

[3] Microsoft Corporation, "Search product lifecycle," Microsoft Corporation, 21 01 2019. [Online]. Available: https://support.microsoft.com/en-us/lifecycle/search?alpha=Microsoft%20Visual%20FoxPro%207.0%20Professional%20Edition. [Accessed 21 01 2019].