

## NUMB3RS Activity: Phreak Out! Episode: "Sacrifice"

**Topic:** Van Eck phreaking

**Grade Level:** 9 - 12

**Objective:** To gain a basic understanding behind the principle of electronic eavesdropping

**Time:** 20 - 30 minutes

**Materials:** TI-83 Plus/TI-84 Plus graphing calculator

### Introduction

While investigating a murder victim's apartment, agent Sinclair finds a small bit of wire dangling from the security alarm control panel. The FBI later determines that the wire was from a high gain antenna, which enabled the victim's estranged wife to electronically eavesdrop on the images on his computer screen. This process is called "van Eck phreaking." While the technical aspects involved are beyond the scope of this activity, students will have the chance to see how screen images are recreated from the electromagnetic radiation (EMR) data collected through phreaking.

### Discuss with Students

All electronic equipment emits electromagnetic radiation (EMR). Cell phones are not permitted on air flights because of a concern that their EMR might disrupt radio communications and navigation equipment. A computer's EMR can carry up to half a mile, even farther if there are conductors like power lines or metal water pipes. Even a handheld calculator's EMR can reach several feet. On a computer monitor, electron beams from the cathode ray tube (CRT) scan from left to right and top to bottom very quickly to create an image. The intensity of these beams determines the appearance of each pixel on the screen. This in turn generates the EMR.

It is possible to use special equipment to "decode" this radiation and recreate the image on the screen. This process is called van Eck phreaking, named after Dutch computer researcher Wim van Eck. In 1985, van Eck wrote a paper describing the process and how to do it; with the right background and equipment, it is not difficult to do (see "Additional Resources"). He used the term "phreak," which previously referred to people who hacked into telephone systems to make free calls. Since before World War II, the military has been studying this and other surveillance techniques. A U.S. government project called TEMPEST deals with this type of electronic eavesdropping.

If possible, students should work in small groups, at least when they get to question 4, which is relatively complicated and has a variety of multi-step strategies for solving.

Note that while some students may find it beneficial to actually key in the PHREAK program on their calculators, the program file can also be downloaded for free by going to <http://education.ti.com/exchange> and searching for "7712."

### **Student Page Answers:**

1. 786,432 2.  $130/530 \approx 24.5\%$ . 3. Answers will vary. Sample response: If the number of wider letters described is assumed to be roughly equal to the number of narrower letters, then this is a reasonable assumption. 4. Answers will vary. A very naïve (and incorrect) estimate can be obtained by ignoring the spaces between lines of text, yielding  $786,432 \times 0.245 \approx 192,676$ . If this arises, discuss its shortcomings with the class. Sample solution: a row of text occupies approximately (from #2) 24.5% of  $(10 \times 1,024)$  pixels  $\approx 2,509$  pixels. Spaces between lines of text use  $8 \times 1,024 = 8,192$  pixels. Because lowercase letters are 7 pixels high, a line of text with space beneath it is 15 pixels high. This means that there is room for  $768 \div 15 \approx 51$  (actually 51.2) lines of text. So, the total number of pixels for the actual text is  $51 \times 2,509 = 127,959$ , or approximately 128,000 pixels. Other approaches are equally feasible. 5. See student work.

Name: \_\_\_\_\_

Date: \_\_\_\_\_

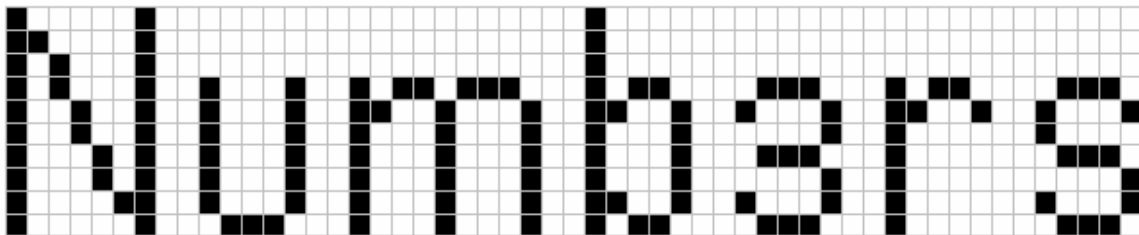
## NUMB3RS Activity: Phreak Out!

While investigating a murder victim's apartment, agent Sinclair finds a small bit of wire dangling from the security alarm control panel. The FBI later determines that the wire was from a high gain antenna, which enabled the victim's estranged wife to electronically eavesdrop on the images on his computer screen. This process is called "van Eck phreaking." Electromagnetic radiation (EMR) is intercepted from a computer's monitor, and then special equipment "decodes" it to recreate the image on the screen. This activity does not involve the process used to generate the EMR data, but it is available in the "Extensions." In this activity you can learn a way to decode the data into the image.

Computer monitors come in many sizes, as well as with different numbers of pixels (picture elements), the small "dots" you see when you look at the screen very closely or with a magnifying glass. The computer monitor on which this activity was written is 1,024 (wide)  $\times$  768 (tall) pixels.

Suppose that in a certain font (style of type), capital letters are 10 pixels high and 7 pixels wide. Lowercase letters are 7 pixels high and 5 wide. Naturally there are some exceptions, like *m*'s, *w*'s, *i*'s, *j*'s, and *l*'s. Generally, it is fairly safe to use 5 pixels as the average letter width. When typing, the spaces between letters are 2 pixels wide. The horizontal space between rows of text, from the bottom of one row of letters (ignoring tails, like on *g*'s and *y*'s) to the tops of the lower case letters in the next line (ignoring capitals and tall letters, like *b*'s and *d*'s), is 8 pixels high.

Here is a representative sample of text, shown highly magnified:



Use the data in the above paragraph and the sample text to answer these questions:

1. How many total pixels are on a computer monitor that measures 1,024  $\times$  768 pixels? \_\_\_\_\_
2. What percent of the pixels in the sample above are black (contain the actual text)? This will be a representative sample of all text. \_\_\_\_\_
3. Give a justification that using 5 as the overall character width in pixels is a reasonable assumption.
4. Suppose that the average word length is five characters. Determine a reasonable estimate for the total number of pixels that contains the text on a full screen of text for the computer described above. Do not ignore the spaces between lines. Explain your reasoning.

On a computer monitor, electron beams from the cathode ray tube (CRT) scan from left to right and top to bottom very quickly to create the image. The intensity of these beams determines the appearance of each pixel on the screen. This in turn generates the EMR. One reason you were asked the previous questions is that in order to recreate the screen image, it is only necessary to know which of the pixels are "lit" (contain the actual text or image on the screen).

Suppose that the "decoding" equipment returns this information to you in the form of coordinate strings; the first number is the row number, followed by the set of vertical coordinates (columns) in that row that are lit. For example, the string (3: 1, 5, 12, 16) means that in row 3, the pixels in columns 1, 5, 12, and 16 are lit. For real-time phreaking, the first number can range from 1 to 768, the columns from 1 to 1,024, and the process would be repeated many times per second (the "refresh" rate).

5. Use these strings to recreate the image on the model screen shown.

- (4: 10, 11, 12, 13, 14, 15, 16)
- (5: 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18)
- (6: 7, 8, 9, 17, 18, 19)
- (7: 6, 7, 19, 20)
- (8: 5, 6, 20, 21)
- (9: 4, 5, 9, 10, 11, 15, 16, 17, 21, 22)
- (10: 4, 5, 9, 10, 11, 15, 16, 17, 21, 22)
- (11: 4, 5, 9, 10, 11, 15, 16, 17, 22, 23)
- (12: 3, 4, 22, 23)
- (13: 3, 4, 22, 23)
- (14: 3, 4, 22, 23)
- (15: 3, 4, 8, 9, 17, 18, 22, 23)
- (16: 4, 5, 9, 10, 16, 17, 21, 22)
- (17: 4, 5, 10, 11, 12, 14, 15, 16, 21, 22)
- (18: 4, 5, 6, 12, 13, 14, 20, 21, 22)
- (19: 5, 6, 7, 19, 20, 21)
- (20: 6, 7, 8, 18, 19, 20)
- (21: 7, 8, 9, 10, 16, 17, 18, 19)
- (22: 9, 10, 11, 12, 13, 14, 15, 16, 17)
- (23: 11, 12, 13, 14, 15)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
10																										
11																										
12																										
13																										
14																										
15																										
16																										
17																										
18																										
19																										
20																										
21																										
22																										
23																										
24																										
25																										

Finally, here's a short program for your graphing calculator that simulates the "refresh" procedure. Each command has a comment with it, explaining what it does.

```
PROGRAM: PHREAK
:For(I,1,5)
:ClrHome
:For(R,1,8)
:For(C,1,16)
:If rand<0.4
:Output(R,C,"X")
:End
:End
:End
```

- Notes:**
- Repeats the "refresh" process 5 times
  - Clears the screen, puts the cursor in the top left corner
  - The screen has 8 rows for text ("for each of the 8 rows")
  - Each row has 16 columns ("for each column in each row")
  - If a random decimal between 0 and 1 is less than 0.4, then...
    - ...an "X" is printed in the current row and column.
  - Closes the "For" loop.
  - Closes the "For" loop.
  - Closes the "For" loop.

*The goal of this activity is to give your students a short and simple snapshot into a very extensive mathematical topic. TI and NCTM encourage you and your students to learn more about this topic using the extensions provided below and through your own independent research.*

## Extensions

### Introduction

Van Eck phreaking raises some interesting mathematical and scientific questions. It also offers opportunities for further study. Originally it was thought that electronic eavesdropping was extremely complicated, requiring an extensive background and very sophisticated equipment. Wim Van Eck made it very clear that it is not only relatively easy to phreak someone's computer but that preventing it is very difficult.

### For the Student

Here are some of the questions you can investigate related to van Eck phreaking and other forms of electronic surveillance:

1. How does a cathode ray tube (CRT) in a computer monitor actually work?
2. What kind of mathematics is used to set up the decoding equipment used for phreaking?
3. What kind of physical devices (like shields) can prevent a computer (or any other electronic device) from emitting the radiation so that it can not be phreaked?
4. What mathematical solutions might be used to prevent a phreaker from being able to reconstruct the images (like scrambling the order in which the image is generated by using some form of cryptography)?
5. What other kinds of electronic eavesdropping are there, such as for cell phones, fax machines, wireless networks, and Internet connections?

Some of these and other questions are discussed in van Eck's original paper in *Computer & Security* called "Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?"

### Additional Resources

- To read Wim Van Eck's original 1985 paper (18 pages, fairly technical as it progresses, but some nice graphics and photographs), see:  
<http://web.archive.org/web/20000830130750/www.shmoo.com/tempest/emr.pdf>
- For a description of the U.S. government project TEMPEST (Transient Electromagnetic Pulse Surveillance Technology), see:  
<http://www.tech-faq.com/tempest.shtml>
- The National Security Agency (NSA) manages the standards and policies for TEMPEST. Although most details are classified, you can see a description of what is called the "TEMPEST Endorsement Program" at:  
<http://www.nsa.gov/ia/industry/tempest.cfm>
- For more information about the programming capabilities of the TI-83 Plus/TI-84 Plus graphing calculators, consult your calculator's guidebook. These guidebooks can be downloaded for free from Texas Instruments at:  
[http://education.ti.com/educationportal/appsdelivery/download/download\\_select\\_product.jsp?cid=us&displaymode=G&contentpaneid=17](http://education.ti.com/educationportal/appsdelivery/download/download_select_product.jsp?cid=us&displaymode=G&contentpaneid=17)